Wireless Networks

Wanqing Guan
Haijun Zhang

Network Slicing for Future Wireless Communication

Theory and Application

Springer

Wireless Networks

Series Editor

Xuemin Sherman Shen, University of Waterloo, Waterloo, ON, Canada

The purpose of Springer's Wireless Networks book series is to establish the state of the art and set the course for future research and development in wireless communication networks. The scope of this series includes not only all aspects of wireless networks (including cellular networks, WiFi, sensor networks, and vehicular networks), but related areas such as cloud computing and big data. The series serves as a central source of references for wireless networks research and development. It aims to publish thorough and cohesive overviews on specific topics in wireless networks, as well as works that are larger in scope than survey articles and that contain more detailed background information. The series also provides coverage of advanced and timely topics worthy of monographs, contributed volumes, textbooks and handbooks.

Wanqing Guan • Haijun Zhang

Network Slicing for Future Wireless Communication

Theory and Application

Springer

Wanqing Guan
Department of Communication Engineering
University of Science and Technology
Beijing
Beijing, China

Haijun Zhang
Department of Communication Engineering
University of Science and Technology
Beijing
Beijing, China

ISSN 2366-1186 ISSN 2366-1445 (electronic)
Wireless Networks
ISBN 978-3-031-58228-8 ISBN 978-3-031-58229-5 (eBook)
https://doi.org/10.1007/978-3-031-58229-5

Contents

Acronyms

5G	The 5th Generation Mobile Networks
6G	The 6th Generation Mobile Networks
3GPP	The 3rd Generation Partner Project
AR/VR	Augmented or Virtual Reality
UAV	Unmanned Aerial Vehicle
MNOs	Mobile Network Operators
FeMBB	Further enhanced Mobile Broadband
umMTC	ultra-massive Machine-type Communications
eURLLC	extremely Ultra-Reliable and Low-Latency Communications
LDHMC	Long-Distance and High-Mobility Communications
ELPC	Extremely Low-Power Communications
NS	Network Slicing
CAPEX/OPEX	Capital Expenditure and Operational Expenditure
VNFs	Virtual Network Functions
RAN	Radio Access Network
E2E	End-to-End
InPs	Infrastructure Network Providers
SDN	Software-Defined Network
NFV	Network Function Virtualization
SPs	Service Providers
MANO	Management and Orchestration
ITU	International Telecommunication Union
eMBB	enhanced Mobile Broadband
mMTC	massive Machine-Type Communications
uRLLC	ultra-Reliable and Low-Latency Communication
OTT	Over-the-top
QoS	Quality of Service
MVNOs	Mobile Virtual Network Operators
QoE	Quality of Experience
NSPs	Network Slice Providers
NSCs	Network Slice Customers

AI	Artificial Intelligence
ML	Machine Learning
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
DNN	Deep Neural Network
DQL	Deep Q-Learning
UDN	Ultra-Dense Network
BSs	Base Stations
NSI	Network Slice Instance
OSS	Operational Support System
BSS	Business Support System
CN	Complex Network
BA	Barabási-Albert
WS	Watts-Strogatz
NW	Newman-Watts
ER	Erdös-Rényi
NPF	Network Slice Policy Function
NSLDB	Network Slice Database
NSSI	Network Slice Subnet Instance
SLA	Service-Level Agreement
VNE	Virtual Network Embedding
BFS	Breadth First Search
KSP	K Shortest Paths
CON	Core Node
OS	Optical Switches
VMs	Virtual Machines
DCs	Data Centers
GRM	Global Resource Manage
LRM	Local Resource Manage
MRP	Markov Renewal Process
MDP	Markov Decision Process
DDQL	Dueling Deep Q-Learning
EUs	End Users
RMO	Resource Management and Orchestration
A3C	Asynchronous Advantage Actor-Critic
DDPG	Deep Deterministic Policy Gradient
MADRL	Multi-Agent Deep Reinforcement Learning
TSN	Time-Sensitive Networking
ULL	Ultra-Low Latency
CT	Communication Technology
OT	Operation Technology
IT	Information Technology
MR	Mixed Reality
SBT	Sporadic Burst Traffic
PTT	Periodic Time-sensitive Traffic

NDT	Non-Deterministic Traffic
TT	Time-Triggered
GCL	Gate Control List
TAS	Time-Aware Shaper
5GS	5G System
IETF	Internet Engineering Task Force
DetNet	Deterministic Networking
PREF	Packet Replication and Elimination Function
AVB	Audio Video Bridging
TG	Task Group
NR	New Radio
CNC	Centralized Network Configuration
CUC	Centralized User Configuration
DSTT	Device-Side TSN Translator
NWTT	Network-side TSN Translator
TN	Transport Network
CU	Central Unit
DU	Distributed Unit
IIoT	Industrial Internet of Things
UPF	User Plane Function
UE	User Equipment
PSFP	Per Stream Filtering and Policing
AF	Application Function
PRBs	Physical Resource Blocks
MAC	Medium Access Control
RNN	Recurrent Neural Network
NWDAF	Network Data Analytics Function
HARQ	Hybrid Automatic Repeat reQuest
SINR	Signal-to-Interference-plus-Noise Ratio
BLER	Block Error Rate

Chapter 1
Introduction

1.1 Development of Network Slicing

Mobile communication system has a profound impact on all aspects of human life, and the demand for higher performance mobile communication has never stopped. The rapid growth of mobile data traffic and the number of mobile terminals brings great challenges to the future wireless network construction [11]. In addition to emerging services such as ultra-high-definition video, virtual reality, and augmented reality that will be supported by smart devices, a variety of new mobile services derived from the development of vertical industries such as intelligent driving and energy Internet will also emerge rapidly. The endless intelligent applications bring huge data flow but also put forward higher performance requirements for the future mobile communication system.

Fifth-generation (5G) networks have been deployed commercially at the end of 2019 and researches on sixth-generation (6G) networks are under way in several countries and organizations [25]. In the coming era of 6G, applications such as augmented or virtual reality (AR/VR), unmanned aerial vehicles (UAV), fully autonomous driving, satellite-ground communications, etc. are forcing mobile network operators (MNOs) to carry the complex scenarios and deliver diverse services [7]. Following these applications, scenarios supported by 6G include further enhanced mobile broadband (FeMBB), ultra-massive machine-type communications (umMTC), extremely ultra-reliable and low-latency communications (eURLLC), long-distance and high-mobility communications (LDHMC), and extremely low-power communications (ELPC) [24], as shown in Fig. 1.1.

As the differentiation features of various wireless services become prominent, operators need to provide network capabilities and resources matching service requirements to improve the quality of user experience. Establishing a private network for each category of service is hard to meet the requirements of a wide range of applications simultaneously while bringing unbearable cost increment to operators. In order to provide various customized services using limited network resources

W. Guan, H. Zhang, *Network Slicing for Future Wireless Communication*, Wireless Networks, https://doi.org/10.1007/978-3-031-58229-5_1

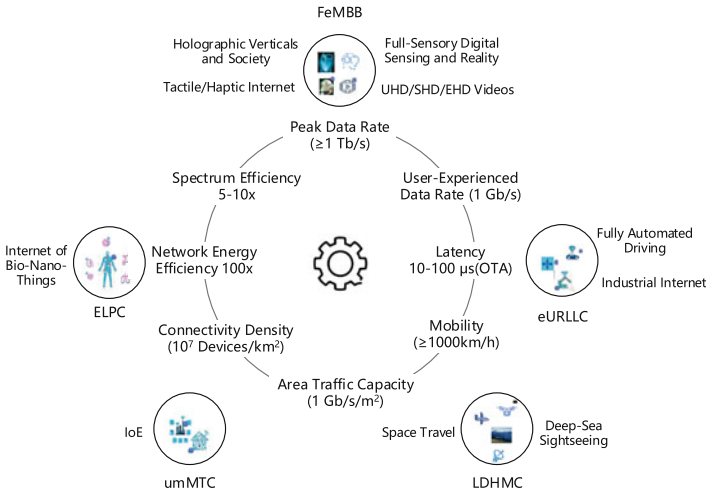


Fig. 1.1 The typical scenarios of 6G networks

of a common infrastructure network, network slicing (NS) has been proposed by wireless industries [12, 15]. Depending on NS technology, differentiated resource requirements could be satisfied flexibly while capital expenditure and operational expenditure (CAPEX/OPEX) could be decreased. As a fundamental attribute of 5G and beyond, NS has been developed rapidly based on the efforts from the industry and academia [8]. Definitely, NS which are introduced and developed in 5G will be inherited and further innovated in 6G.

1.2 Basic Concept and Technologies

In order to cope with the trend of differentiation and customization of mobile services and improve the quality of service experience of end users, operators need to abandon the deployment and construction of traditional mobile networks. By creating a dedicated, virtualized, and isolated logical network for services with strictly different requirements in a common infrastructure network, operators can optimize network resource usage efficiency while saving costs. This new construction idea is the typical application of network slicing concept which is the most concerned in the field of wireless network research. Based on virtualization technology, network slicing divides the infrastructure network into multiple logical networks to meet the differentiated and customized service requirements of vertical industries.

The core of network slicing is the implementation process of a network slicing instance. An instance consists of multiple virtual network functions (VNFs) and computing, storage, and network resources across multiple technical domains,

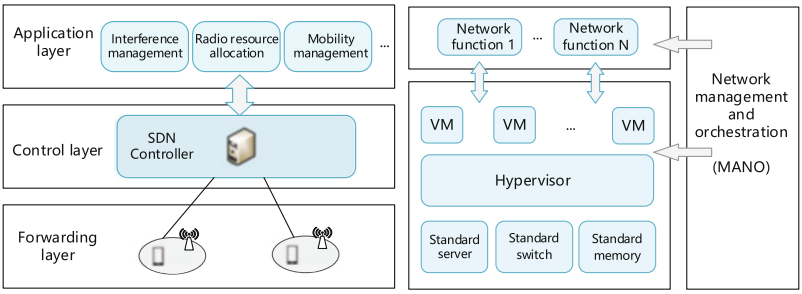
including radio access network (RAN), transport network, core network, and data centers that manage third-party applications in industry verticals. The implementation of network slicing is based on the following three important principles:

- **Isolation.** Slices must be isolated from each other, so that congestion or failure on one slice will not affect other slices. However, isolation may come at the cost of reduced multiplexing gain, which results in lower network resource utilization.
- **Customization.** Resources allocated to a specific tenant must be effectively utilized to meet the corresponding service requirements to the maximum extent.
- **End-to-end (E2E).** It includes two aspects: (i) slices need to span different administrative domains, which means that a slice occupies heterogeneous resources provided by different infrastructure network providers (InPs); (ii) slices need to across different technology domains, including RAN domain, transport network domain, and core network domain.

1.2.1 Enabling Technologies of Network Slicing

A network slice is defined by International Telecommunication Union (ITU) as a logically isolated network partition consisting of multiple VNFs, which is isolated and equipped with a programmable control plane and a data plane [10]. The VNFs that constitute the network slice may vary greatly according to specific service requirements. The service types associated with the network slice determine the resources allocated to this network slice and the corresponding processing flow. The realization of network slicing is inseparable from software defined network (SDN) architecture and network function virtualization (NFV) technology [8]. SDN architecture is an appropriate technology for the configuration and control of the forwarding planes of the underlying resources, while NFV can manage the life cycle of network slices and orchestrate VNFs efficiency. As the key technologies to realize network slicing, SDN and NFV enable network components to run on virtualized infrastructure networks in the form of software and provide virtual resources according to the requirements of slices.

The core of SDN concept is to separate the control plane of network equipment from the data plane, with which network management applications are no longer dependent on hardware devices, and can be flexibly developed and loaded according to the actual network requirements. NFV enables network functional components to run as software, decouples them from the infrastructure on which they run and provides the flexibility to instantiate and assign network functions anywhere in the network or data center. The comparison diagram of SDN and NFV architecture is shown in Fig. 1.2. Network slicing technology combines the flexible characteristics of network resources virtualization and the open characteristics of SDN, which solves the problem of rigid traditional network management architecture, and at the same time opens the network management ability to adapt to the increasingly large and complicated wireless network.



(a) SDN architecture (b) NFV architecture

Fig. 1.2 The comparison diagram of SDN and NFV architecture

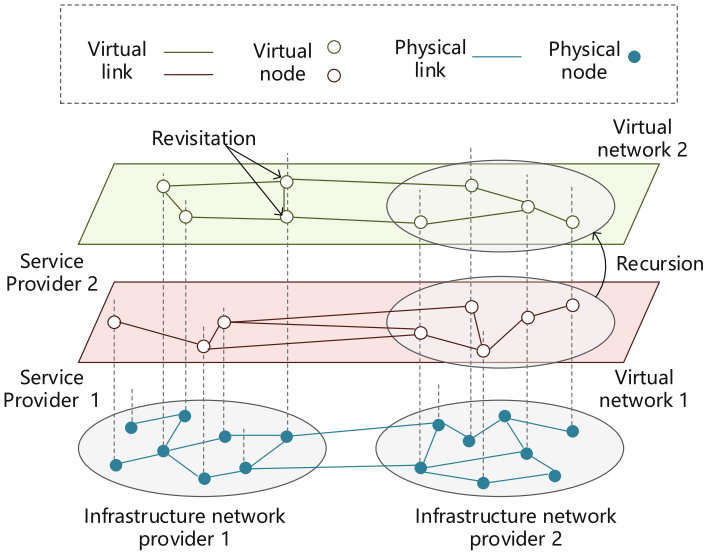


Fig. 1.3 The environment of network virtualization

1.2.2 Implementation of Network Slicing

Partitioning a network into multiple logical subnetworks based on virtualization technology is not a new concept, which has long been studied in the network field. Network virtualization provides flexibility, promotes diversity, and ensures security and manageability by allowing multiple heterogeneous network architectures to coexist as a shared physical infrastructure [14]. Network slicing and network virtualization have similar service models and service roles. The relationship between these roles in the environment of network virtualization is shown in Fig. 1.3.

Network virtualization aims to separate the role of traditional Internet service providers (SPs) into two separate entities, the InPs which manage the physical

infrastructures, and SPs which create virtual networks by aggregating resources from multiple InPs to provide E2E services. InPs manage resources of infrastructure network and provide these resources to different SPs through programmable interfaces. SPs rent resources from multiple InPs to create and deploy virtual networks and provide E2E services to end users by dividing network resources. A single SP can also act as an InP to provide sources for other SPs, which is known as recursion. Multiple virtual nodes in a single virtual network can be deployed on the same physical node at the same time, which is called revisitation.

Based on the above business model and role relationship, the general process of network slicing is as follows:

(1) Multiple tenants define slice templates according to the application scenarios and service demands.
(2) InPs design slice instances based on slice templates.
(3) The management and orchestration (MANO) function allocates resources to slices based on their requirements.
(4) The slice instances are deployed and implemented.
(5) The running slices are monitored and managed, which contributes to adjusting resources in response to the change in requirements.

As the enabling technology for the above processes, SDN/NFV enables various network functions to run on virtual machines. Different types of network slices combine these VNFs based on service requirements to logically form multiple private networks isolated from each other. In conclusion, the rise of the concept of network slicing makes the mobile network more flexible and open, and SDN/ NFV technology gives users the ability of simple and quick customization on demand. The common physical infrastructure network is divided into multiple proprietary logical networks that match service requirements. On the one hand, operators can enrich their profit modes, reduce deployment costs and improve operation efficiency. On the other hand, it opens up network capabilities to provide third-party tenants with higher quality of service experience and finally achieve a win-win situation.

In network slicing based 5G system, resources of multi-domain infrastructure network can be efficiently allocated to multiple network slices according to the requirements of use cases [23]. As shown in Fig. 1.4, the set of VNFs within the physical infrastructure is logically separated to build dedicated logical networks according to the use case families. The ITU has identified three broad use case families [17]: enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable and low-latency communication (uRLLC). A network slice is composed of a collection of VNFs and specific access technology, and the composition depends on the characteristics of use case.

Not all slices contain the same set of VNFs, and some slices may even be missing some VNFs that are critical to mobile networks. VNFs that compose different types of network slices are deployed on multiple physical servers in the same infrastructure network. Therefore, the virtual node set corresponding to the VNFs set of slices is a subset of the physical node set corresponding to the set of physical servers. Moreover, since the slices are dynamically created based on the service

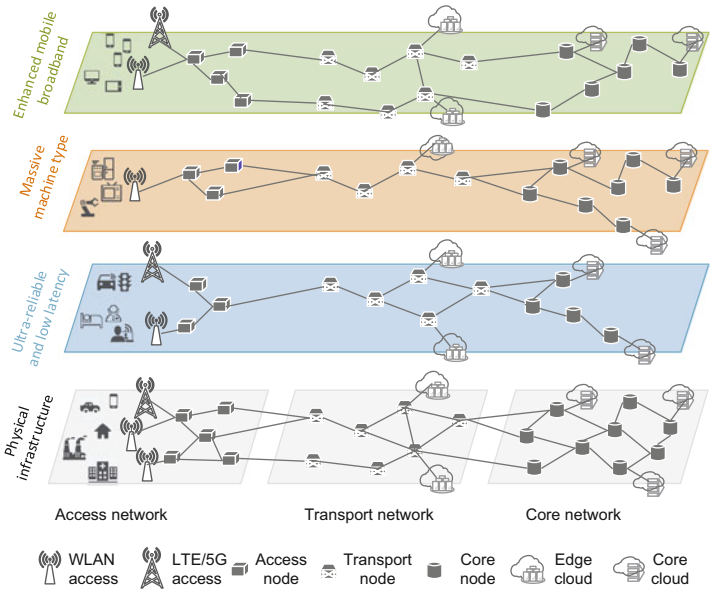


Fig. 1.4 5G network slicing architecture

requirements corresponding to the use cases, the service requirements should be expressed accurately and mapped flexibly to the set of VNFs. Differences in service requirements determine the differences in the structure characteristics of these slice. By comparing the diagram of 5G network slicing architecture with the diagram of network virtualization environment introduced previously, it can be seen that the deployment of network slice on the underlying infrastructure network is the same as the deployment of virtual network on the infrastructure network, including the deployment of virtual nodes and virtual links.

1.3 Demand for Future Development

As a key innovation expected to be inherited in 6G, network slicing is able to reduce CAPEX/OPEX by sharing the network resources among multiple tenants. Tenants, such as mobile virtual network operators (MVNOs), over-the-top (OTT) and vertical industries with limited capacity or coverage, rent the physical resources of MNOs or InPs to provide diversified services. To further reduce CAPEX/OPEX and increase revenue opportunities, tenants are motivated to unite the available resources provided by different InPs to enhance their attractiveness and acquire more subscribers [16]. Therefore, there is a tremendous need to efficiently manage multi-dimensional resources of multi-InPs while meeting the strict and diversified service requirements of multi-tenant under dynamic environment.

1.3.1 Multi-Dimensional Resource Management for Multi-InPs

The introduction of network slicing has undoubtedly brought many advantages to wireless networks and facilitated the implementation of business applications in vertical industries. However, the future development of network slicing still faces various challenges. The maturity level of researches on various aspects of network slicing is given in [8]. It is pointed out that although the enabling technology for network slicing has been mature enough, the related aspects of E2E network slicing have not been well understood and fully solved. As for the future wireless network, both the number and scale of network slices are bound to be huge. The rapidity of slice deployment and management will directly affect the promotion speed of new services and the grasp of business opportunities. Achieving rapid deployment and efficient management of E2E slices while ensuring quality of service (QoS) and the isolation among slices still be the focus of future research in the field of network slicing.

In the evolution process of wireless network, the explosive data traffic makes it urgent to improve network capacity, which results in dense deployment of base stations with small coverage, low power and low cost [5]. Also, the ubiquitous access makes ultra-dense deployment gradually become the evolution trend of radio access network. However, this dense deployment requires a revolutionary upgrade of the existing network to provide QoS assurance for various services. The cost of such upgrades is prohibitive for most MNOs, and near-limit network densities incentivize MNOs to share their network infrastructure and available resources through technologies such as dynamic spectrum management. Therefore, the traditional business model in which several large MNOs achieve E2E service provision through independent deployment and expansion of infrastructure is approaching the limit point, the change of business model is imminent. As shown in Fig. 1.5, business models that encourage collaboration between MNOs and other market participants have emerged.

Wireless networks need to adopt new partnerships and business models for different types of customers, which becomes a key asset underpinning verticals [2]. Here are five roles that have attracted attention in the new collaborative business model:

- **InPs**, which are responsible for providing physical resources and infrastructure maintenance and update, while MNOs that interact with other MNOs but not directly with end users can also play the role of InPs.
- **Cloud InPs**, which provides computing and storage resources and potential cloud services to third parties, including some platform services, such as Openstack provided by Linux and Elastic Compute Cloud provided by Amazon Web Services, Kubernetes provided by Google and Azure provided by Microsoft.
- **MVNOs**, which lack network infrastructure and have limited capacity or coverage lease physical resources from existing MNOs/InPs.

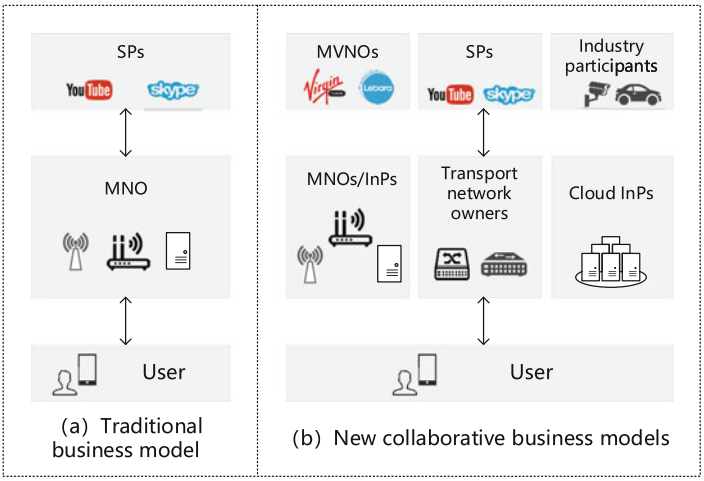


Fig. 1.5 The schematic diagram of traditional business model versus new collaborative business model

- **SPs**, which provides services that run at optimal performance on the network belonging to the MNOs. High-data-consuming applications may prompt application providers, such as Netflix and Hulu, to buy network resources from MNOs to encourage end users to use their services.
- **Vertical market participants**, which leverage the network and cloud resources provided by MNOs/InPs and cloud InPs to provide multiple services to non-telecom specific industries, such as factories, transportation, and healthcare.

To extend coverage or capacity, multiple MNOs can combine their networks into a joint InP, leveraging hierarchical orchestration for real-time flexible resource management and sharing [1]. Based on the network sharing architecture proposed by 3GPP Release 14 [1], the authors in [22] proposed a network sharing architecture suitable for multi-tenant slicing. Based on the resource pooling technology, a group of co-existing MNOs is combined into a federated InP, and the resources of this federated InP is segmented to create a slice to provide E2E services for multi-tenants. Although this architecture provides a solution to realize the cooperation and sharing between MNOs/InPs, the cooperation process is still faced with the problem of cooperation game between each other. The choice faced by MNOs is whether to deploy wireless network independently or form joint InP through resource pooling, which depends on the economic benefits obtained by these MNOs, and this economic benefit is reflected in whether joint InP can meet the service quality requirements of multi-tenant slicing.

It is clear that the collaboration between MNOs needs to satisfy the requirements of end-to-end slices deployed on the collaboration network, taking into account the structural characteristics of each infrastructure network. Such collaboration can promote sustainable revenue for wireless networks and help operators overcome the

dilemma of cost-revenue mismatch. Therefore, the research of collaboration strategy among multiple infrastructure networks is also of great significance, which lays a foundation for efficient management and quality assurance of end-to-end network slices. In addition, slices allocated to verticals can stretch across multiple administrative domains and occupy resources from different MNOs. A multi-domain network slicing orchestration architecture and federated resource management are required to address the challenges in multi-domain slice instantiation [20].

1.3.2 Dynamic Orchestration for Multi-Tenant Slicing

For services in 6G scenarios, guaranteeing extreme quality of experience (QoE) continuously requires rapid adjustment of network parameters based on real-time monitoring of network status. In order to guarantee QoE and boost revenue, efficient sharing of the underlying infrastructure has stimulated the interest of the research community [4]. By establishing efficient network sharing schemes, multiple tenants which may own conflicting resource requirements obtain access to the different parts of the limited resources. As service providers, tenants rent resources to offer slice instances according to heterogeneous service requirements, which enhances the existent resource sharing flexibility.

Creating customized slices for multiple tenants according to their preferences enables flexible and adaptive resource management [23]. Moreover, allowing tenants to customize the resource allocation for each slice can dynamically adapt to the changes in network environment caused by user mobility, time-varying channel conditions and so on. However, supporting more innovative services and satisfying increasingly-diverse user demands impose significant challenges for multi-tenant slicing, particularly in terms of dynamic slice orchestration.

The first challenge is how to achieve real-time status observation of slices by depicting dynamic slice deployment and scalable resource utilization. Slice status information should be accurately obtained and quickly incorporated in decision-making of resource allocation. Then, efficient resources planning is conducted based on current status information of slices, including reserving resources for slices and determining the placement of VNFs for differentiated slices.

Considering that an E2E slice consists of a number of interconnected VNFs from RAN, core network, and transport network, combinatorial optimization of numerous resources is the second challenge. The differences in profit of providing multiple resources to different tenants need to be accounted for when maximizing long-term revenue of InPs as network slice providers (NSPs). Striking a balance between the resources utilization of infrastructures and the profits of differentiated services provisioning is crucial for NSPs.

Last but not least, quickly satisfying the dynamic demands of differentiated services is another challenge. Since the scale and rates of network flows keep changing, the resources allocated to slices need to be adjusted in time to cope with the dynamic user demands. Additionally, the growing number and types of slices

result in high complexity of slice adaption. Trading off the cost of reconfiguring slices and the satisfaction of stringent service quality becomes harder for tenants as network slice customers (NSCs).

Artificial intelligence (AI) saw rapid development during the past ten years and solved many pain points in different industries, such as healthcare, autonomous driving, smart manufacturing, etc. As one of the most promising AI tools, machine learning (ML) techniques have been widely applied in wireless communications [19]. Due to the uncertainty of service requirements, many model-free AI-based solutions are applied to jointly allocate multi-dimensional resources to slices [6]. Because resource allocation in wireless networks affects the QoE of services, various resource allocation methods have been studied over the past decades, including optimization, heuristic and game theoretic. As wireless networks become more complex, the static model-based algorithms will be inapplicable in the real dynamic network because of the long decision-making time and high computing burden.

By iteratively learning from the reward feedback of environment, an optimal decision can be quickly achieved with ML methods compared to the conventional model-based optimization methods. Many researches adopt reinforcement learning (RL) based approach to manage resources involving both radio access part and core network part [13]. Owing to the capability of learning an optimal policy quickly, RL has been preferred for decision-making in the time-varying network environments and widely applied in solving many resource management problems, for instance, power control, spectrum management and computation resource management [19]. RL incorporates farsighted system evolution into its decision-making and updates decision strategies to reach optimal performance through feedback of the previous decisions. Moreover, RL has become an effective method to solve the decision-making problem of network slicing in the uncertain and probabilistic environment [3]. As one of the most commonly adopted conventional RL algorithm, Q-learning suffers from slow convergence speed when the state space and action space are large. Deep reinforcement learning (DRL) algorithm which integrates deep neural network (DNN) with RL has been proposed by Google DeepMind, and the application of many advanced DRL algorithms has triggered tremendous research attention [9].

Based on deep Q-network, DRL as shown in Fig. 1.6 outperforms conventional RL because experience replay is used to increase the efficiency of learning and enhance the stability of DNN. After performing action selection, reward calculation and new state observation, the mini-batches of experience are sampled uniformly at random to feed into the neural network during the learning process. DNN which is used to approximate the Q-value function takes the current states as the input and outputs a set of Q-values for all of the state-action pairs. Instead of using Q-table to store Q-values in the Q-learning algorithm, the deep convolutional network is used to address the instability caused by the correlations. Experience replay memory randomizes over the data, thereby allowing for greater efficiency and breaking the strong correlations between the samples. Hence, DNN improves the convergence of Q-learning and enables the deep Q-learning (DQL) algorithm to solve the problems which have a high-dimensional state-action space.

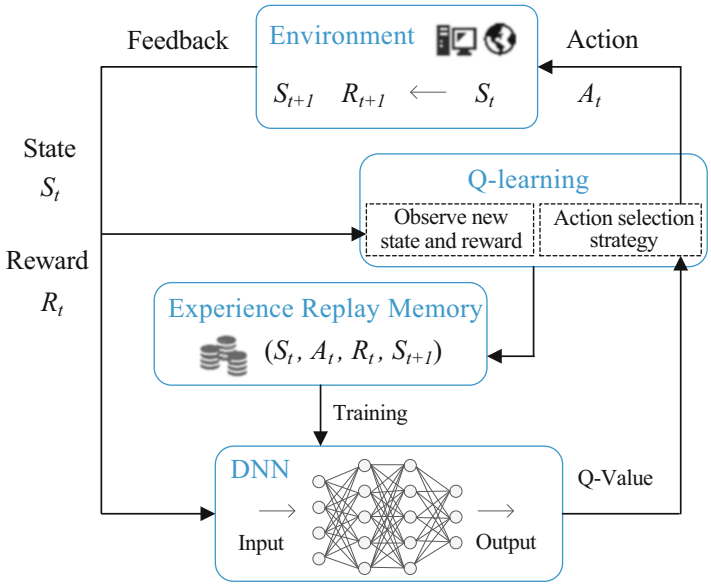


Fig. 1.6 An illustration of deep Q-learning

For each tenant, when the traffic flow arrival/departure results in the degradation of slice's service quality, individually deciding how to reallocate available resources is necessary. Owing that the traffic variation cannot be predicted without error, RL methods are also need to be applied in slice adaption decisions of slice orchestration architecture. However, the existing RL based resource allocation methods [13, 21] and the AI-assisted network architecture for network slicing [18] are inadequate to balance the capability of customizing resources for multiple tenants and maximizing revenues for multiple InPs. It is still very challenging to allocate resources across multiple domains and customize resources for each tenant simultaneously.

Acknowledgments If you want to include acknowledgments of assistance and the like at the end of an individual chapter please use the `acknowledgement` environment—it will automatically render Springer's preferred layout.

References

1. 3GPP: Telecommunication management; network sharing; concepts and requirements (2016)
2. 5GPPP: 5G empowering vertical industries (2016)
3. Abiko, Y., Saito, T., Ikeda, D., Ohta, K., Mizuno, T., Mineno, H.: Flexible resource block allocation to multiple slices for radio access network slicing using deep reinforcement learning. IEEE Access **8**, 68183–68198 (2020)

4. Antonopoulos, A.: Bankruptcy problem in network sharing: Fundamentals, applications and challenges. IEEE Wireless Commun. **27**(4), 81–87 (2020)
5. Bhushan, N., Li, J., Malladi, D., Gilmore, R., Brenner, D., Damnjanovic, A., Sukhavasi, R., Patel, C., Geirhofer, S.: Network densification: the dominant theme for wireless evolution into 5G. IEEE Commun. Mag. **52**(2), 82–89 (2014)
6. Chen, X., Zhao, Z., Wu, C., Bennis, M., Liu, H., Ji, Y., Zhang, H.: Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach. IEEE J. Sel. Areas Commun. **37**(10), 2377–2392 (2019)
7. David, K., Berndt, H.: 6G vision and requirements: Is there any need for beyond 5G? IEEE Veh. Technol. Mag. **13**(3), 72–80 (2018)
8. Foukas, X., Patounas, G., Elmokashfi, A., Marina, M.K.: Network slicing in 5G: Survey and challenges. IEEE Commun. Mag. **55**(5), 94–100 (2017). https://doi.org/10.1109/MCOM.2017.1600951
9. Hua, Y., Li, R., Zhao, Z., Chen, X., Zhang, H.: Gan-powered deep distributional reinforcement learning for resource management in network slicing. IEEE J. Sel. Areas Commun. **38**(2), 334–349 (2020)
10. ITU-T: Framework of network virtualization for future networks, next generation network - future networks (2012)
11. ITU: The forecast of mobile phone flow development before 2030. Tech. rep. (2015)
12. Jiang, M., Condoluci, M., Mahmoodi, T.: Network slicing management & prioritization in 5G mobile systems. In: European Wireless, pp. 1–6 (2016)
13. Li, R., Zhao, Z., Sun, Q., I, C., Yang, C., Chen, X., Zhao, M., Zhang, H.: Deep reinforcement learning for resource management in network slicing. IEEE Access **6**, 74429–74441 (2018)
14. Mosharaf Kabir Chowdhury, N.M., Boutaba, R.: Network virtualization: state of the art and research challenges. IEEE Commun. Mag. **47**(7), 20–26 (2009)
15. Rost, P., Banchs, A., Berberana, I., Breitbach, M., Doll, M., Droste, H., Mannweiler, C., Puente, M.A., Samdanis, K., Sayadi, B.: Mobile network architecture evolution toward 5G. IEEE Commun. Mag. **54**(5), 84–91 (2016)
16. Samdanis, K., Costa-Perez, X., Sciancalepore, V.: From network sharing to multi-tenancy: The 5G network slice broker. IEEE Commun. Mag. **54**(7), 32–39 (2016)
17. Series, M.: IMT vision–framework and overall objectives of the future development of IMT for 2020 and beyond. Recommend. ITU **2083**, 0 (2015)
18. Shen, X., Gao, J., Wu, W., Lyu, K., Li, M., Zhuang, W., Li, X., Rao, J.: AI-assisted network-slicing based next-generation wireless networks. IEEE Open J. Veh. Technol. **1**, 45–66 (2020)
19. Sun, Y., Peng, M., Zhou, Y., Huang, Y., Mao, S.: Application of machine learning in wireless networks: Key techniques and open issues. IEEE Commun. Surv. Tutorials **21**(4), 3072–3108 (2019)
20. Taleb, T., Afolabi, I., Samdanis, K., Yousaf, F.Z.: On multi-domain network slicing orchestration architecture and federated resource control. IEEE Network **33**(5), 242–252 (2019)
21. Van Huynh, N., Thai Hoang, D., Nguyen, D.N., Dutkiewicz, E.: Optimal and fast real-time resource slicing with deep dueling neural networks. IEEE J. Sel. Areas Commun. **37**(6), 1455–1470 (2019)
22. Vincenzi, M., Antonopoulos, A., Kartsakli, E., Vardakas, J., Alonso, L., Verikoukis, C.: Multi-tenant slicing for spectrum management on the road to 5G. IEEE Wireless Commun. **24**(5), 118–125 (2017). https://doi.org/10.1109/MWC.2017.1700138
23. Zhang, H., Liu, N., Chu, X., Long, K., Aghvami, A.H., Leung, V.C.M.: Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges. IEEE Commun. Mag. **55**(8), 138–145 (2017). https://doi.org/10.1109/MCOM.2017.1600940
24. Zhang, Z., Xiao, Y., Ma, Z., Xiao, M., Ding, Z., Lei, X., Karagiannidis, G.K., Fan, P.: 6G wireless networks: Vision, requirements, architecture, and key technologies. IEEE Veh. Technol. Mag. **14**(3), 28–41 (2019)
25. Zong, B., Fan, C., Wang, X., Duan, X., Wang, B., Wang, J.: 6G technologies: Key drivers, core requirements, system architectures, and enabling technologies. IEEE Veh. Technol. Mag. **14**(3), 18–27 (2019)

Chapter 2
Efficient Management of Physical Infrastructure Networks

2.1 Cooperation Among Multiple Infrastructure Operators

The existing management architecture of network slicing mainly focuses on the core network, combining SDN and NFV technologies to implement core network slices [34]. However, ultra-dense networks (UDN) with dense small cells deployed at the access network side is becoming one of the development trends of wireless networks. How to improve the resource utilization at the access side in such dense deployment scenario has attracted much attention [44]. Based on the concept of flexible access network, the authors in [14] propose a fully programmable network slicing architecture characterized by RAN abstraction. This architecture can adjust the resource allocation strategy according to the requirements of RAN slices, which enables flexible and dynamic sharing of wireless resources, such as multiple RAN slices can share base stations (BSs) and antennas. With the maturity of core network slicing technology and the rapid development of RAN slicing, it is increasingly urgent to deploy and manage E2E network slices.

In order to meet the requirements of various E2E services, the management architecture of network slicing needs to consider the deployment and implementation of E2E slices. Figure 2.1 shows the schematic diagram of E2E network slicing implementation. In the E2E management architecture of network slicing, users do not need to consider the selection of core network slices after selecting the RAN slices to be accessed. The management architecture provides E2E slices to select the RAN slices and the matching core network slices for the corresponding services. This E2E slicing technology covering access network, transmission network, and core network domain can more comprehensively meet service requirements, help to achieve rapid deployment of slicing and global on-demand resource scheduling, greatly shorten service response time, and effectively improve the quality of user service experience.

In addition, wireless networks need to support vertical markets by catering to different types of new services. In order to reduce the cost and increase the

W. Guan, H. Zhang, *Network Slicing for Future Wireless Communication*, Wireless Networks, https://doi.org/10.1007/978-3-031-58229-5_2

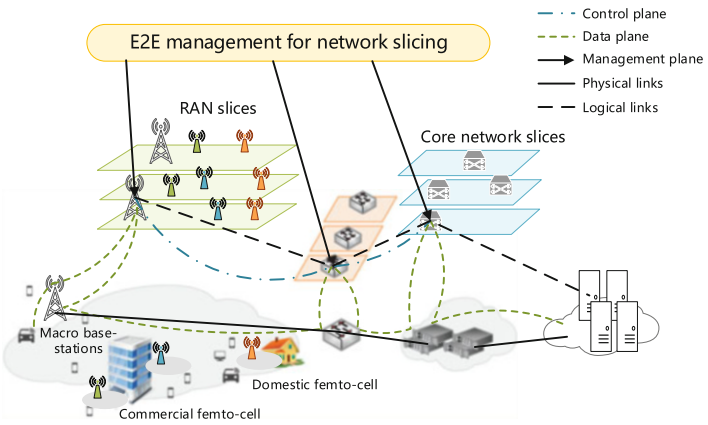


Fig. 2.1 The schematic diagram of E2E network slicing implementation

revenue, multiple MNOs seek cooperation to build a joint infrastructure network. The implementation of network slicing will no longer be limited to a single infrastructure network which belongs to a single MNO. Existing network slicing management architectures, such as the NESMO framework [11], can automate network slicing design, deployment, configuration, activation, and lifecycle management across multiple domains. However, this framework is not suitable for dense deployment scenarios and does not consider collaboration among multiple infrastructure networks. Before creating a collaborative management architecture for network slicing, the requirements for network slicing management need to understood. According to the requirements of different industries and standards associations for network slicing, network slicing management requirements can be classified into the following aspects [11]:

- Flexibility, which enables a MNO to dynamically manage and orchestrate the VNFs and virtual resources of network slices. Each network slice consists of a set of network functions and the resources to support the operation, decision-making, and configuration of these network functions. The network slicing management architecture should support E2E slice management to satisfy distinct requirements which may come from different vendors, carriers, or third parties.
- Customization, which enables a MNO to create, operate, and manage network slices that support a variety of customized end-user services with different requirements for network characteristics in terms of mobility, billing, security, policy control, latency, and reliability. The slice management architecture should support the participation of multiple third parties with similar network characteristics.
- Simplification, which simplifies the logical architecture of mobile networks and reduces the complexity of network slicing operations. Considering that the requirement of flexibility adds complexity to the design and operation of network

slicing, it is difficult to balance the complexity and simplicity to avoid increasing costs.
- Exposure, which exposes network capabilities through an open application program interface, allowing third parties to create and manage network slices within the boundaries defined by MNOs.
- Elasticity, which supports the elasticity of the network slice instance (NSI) in terms of capacity with minimal impact on the service of this slice or other slices.
- Cloudification, which supports cloud computing technologies in network slice deployment.
- Legacy support, which minimizes dependency between the network slicing management solution and Operational Support System/Business Support System (OSS/BSS) in order to support the legacy OSS/BSS systems.
- Lifecycle management, which implements network slice lifecycle management and modifications to a network slice with a minimal impact to active subscriber services.
- Automation, which supports an automated network slice design and deployment from a business service order.
- Isolation, which avoids a fault or high resource usage of a slice to harm the stability or performance of other slices.
- Multi-domain, which supports multi-domain architectures.

For MNOs, there is a need to exploit new revenue resources without significantly increasing capital expenditure and operational expenditure (CAPEX/OPEX). The authors of [36] provide an overview of 3GPP standard evolution from network sharing to multi-tenant systems. They propose a multi-tenant network architecture to enable MVNOs and industry vertical market players to request and lease resources from InPs dynamically. In these researches, different slices are deployed and managed on a single infrastructure network. In order to break the traditional business model of a single infrastructure network ownership, MNOs are motivated to share their infrastructure networks and the available resources [41]. Thanks to network sharing strategies [2, 9], the potential energy and financial benefits spur MNOs in multi-operator environments to participate in the infrastructure network sharing. However, these strategies always focus on the spotlight of cellular networks considering the peak traffic demands. As a result, joint resources allocation among different InPs occurs on a small number of BSs and enables a large number of remaining unused infrastructure. Hence, for a joint venture InP formed by a set of MNOs, resources sharing need to be handled by a centralized network manager.

Moreover, BSs are densely deployed for UDN, which requires a global vision for flexible management among BSs and efficient cooperation among infrastructures. A network slicing management framework for 5G UDN which provides a global view of resource management has been presented by us in [18]. However, the framework is suitable for the scenario of single infrastructure network. In order to reduce the CAPEX/OPEX in multi-tenant slicing, a cooperation strategy among multiple infrastructure networks is urged. The authors of [41] propose a novel scheme defined according to coalitional game theory for cooperation among different MNOs.

However, their cooperation strategy focus on optimizing energy efficiency and do not take into account the demand of different slices and the structural characteristics of InPs. As far as we know, few researches have provided an efficient cooperation strategy for InPs to meet the QoS requirements of slices while considering the differences of infrastructure networks.

2.1.1 Topological Characteristic Analysis

In [19], we investigate cooperation among multiple infrastructure networks for multi-tenant slicing and propose an on-demand cooperation strategy from a complex network (CN) theory perspective. CN theory is usually used to analyze structural characteristics and predict dynamical behaviors of networked systems [6]. The study of CN theory has provided many measures of topological properties and effective ways to model the real-world networks. Here, CN theory is used to obtain the topological information of multiple infrastructure networks. Empirical studies have demonstrated that many real-life communication networks exhibit small-world and scale-free topological properties [27, 43]. Using CN theory to obtain the structural property and analyze the impact of topology on cooperation is reasonable and reliable.

Hence, extensive simulations are performed on four typical complex network topologies to analyze the cooperation among different infrastructure networks. By considering the demand of slices and the structural differences among multiple infrastructure networks, the proposed cooperation strategy sets the rules for collaboration among the InPs in multi-tenant systems. Compared to the existing network slicing models, the on-demand cooperation allows InPs to significantly reduce costs independently while satisfying the demands of slices on the joint infrastructure network.

In order to understand and predict the behavior of real-world networked systems such as the Internet, social networks, and biological networks, researchers of CN theory developed a variety of techniques and models [31]. Considering that many real-world networked systems can be modeled as sets of interconnected networks or networks with multiple types of connections, multiple complex networks as multilayer networks have attracted growing attention [5, 28]. To analyzed the impact of topological characteristic, infrastructure networks are represented by typical network models and the establishment of cooperative relationship is denoted by the creation of interconnections. According to the demand of reducing delay, betweenness centrality of nodes is introduced to obtain the topological information of infrastructure networks. The betweenness centrality quantifies how much a node is found between the path linking other pair of nodes. The betweenness centrality is defined as the fraction of shortest paths between any pair of nodes that travel through the node, which can be denoted by

$$b_n = \sum_{s \neq n \neq t} \frac{\sigma_{st}(n)}{\sigma_{st}}. \tag{2.1}$$

In this equation, σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(n)$ is the number of those paths that pass through node n.

In multi-tenant systems, infrastructure networks spanning both RAN and the core network provide physical resources to support various use cases. For network slicing architecture, there is a MANO entity that translates use cases and service models into network slices by chaining network functions, mapping them to infrastructure resources, and configuring and monitoring each slice during its life cycle [15]. Since multiple InPs seeking coverage/capacity extension pool their networks into a joint venture InP, MANO is required to provide optimal cooperation approaches in order to meet the requirements of use cases. The optimal cooperation approaches determine where the sharing interactions should be created.

Figure 2.2 shows the cooperation structure of infrastructure networks for multi-tenant slicing. There are two InPs that own different networks, respectively, including RAN, core network, transport network, and part of cloud infrastructures. At first, tenants define the demand of slices and send it to NS manager and orchestrator. According to the demand, manager and orchestrator deploy slices which are the chains of VNFs in the joint infrastructure network. Considering that the deployment of slices is effected by the topology of infrastructure network, the creation of the interconnections should combine the demand of slices with the topology information. To effectively demonstrate the validation of on-demand cooperation, the demand of delay-critical slices, as the most versatile demand, is utilized in our simulations.

Autonomous vehicle application, as a typical use case which requires extremely low latency, is used as an example scenario. For an autonomous vehicle to operate safely and effectively, the E2E latency perceived by the end user need to be minimized. E2E latency measures the duration between the transmission of a data packet from the source node to the destination node in the joint infrastructure network. Hence, in order to satisfy the delay-critical demand defined by tenants, the cooperation among infrastructure networks should effectively improve the number of the shortest paths in the joint infrastructure network and reduce the length of shortest paths between node pairs. Improving the number of the shortest paths provides more paths for packets to avoid congestion and reducing the length of shortest paths provides a decrease in transmission time.

Since infrastructure networks provided by InPs have different topologies, we investigate the cooperation among infrastructure networks which are represented by different network models. Here note that the actual characteristics of infrastructure network topologies are not uniform and monotonous as the use cases for multi-tenant slicing are various. Thus, considering the applicability and generality of our cooperation approach, four typical kinds of network topologies are chosen as simulation topologies.

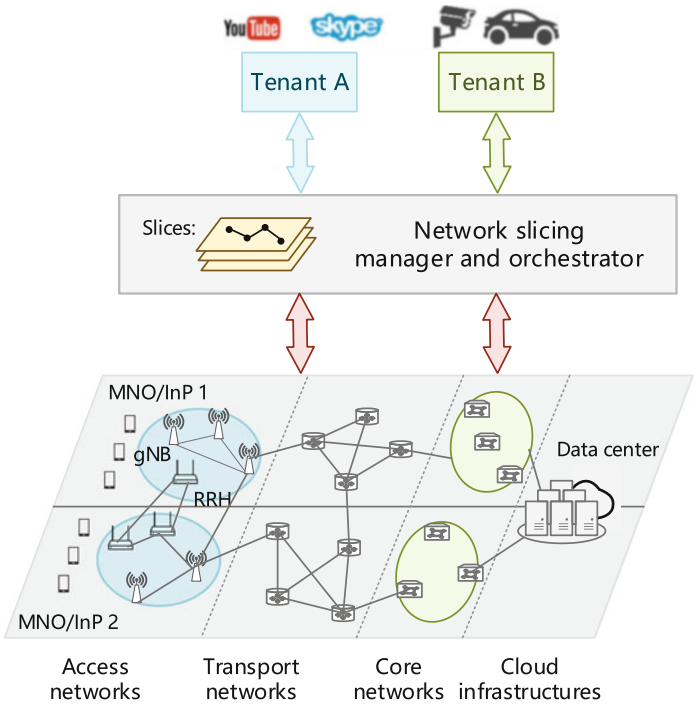


Fig. 2.2 The cooperation structure of infrastructure networks for multi-tenant slicing

These four typical network models are Barabási-Albert (BA) scale-free network model [3], Watts-Strogatz (WS) small-world network model [42], Newman-Watts (NW) small-world network model [32], Erdös-Rényi (ER) random graph network model [7, 13]. Figure 2.3 shows an example of these four typical network models. The BA model, as an evolving network model, aims to reproduce the growth processes taking place in real networks based on two basic ingredients: growth and preferential attachment. The WS model is a method to construct graphs having both the small-world property and a high clustering coefficient. Since the WS algorithm may destroy the network connectivity during the rewiring process, NW algorithm modified the process from rewiring to adding. Hence, NW model has a higher edge density than WS model. The random network refers to the disordered nature of the arrangement of links between different nodes. The ER model can be extended in a variety of ways to make random graphs a better representation of real networks. Their generating principles are introduced as following.

The BA modeling algorithm is as follows:

- Growth: Starting from a connected network of small size N_0, introduce one new node to the existing network each time, and this new node is connected to ne existing nodes in the network simultaneously, where $1 \leq ne \leq N_0$.

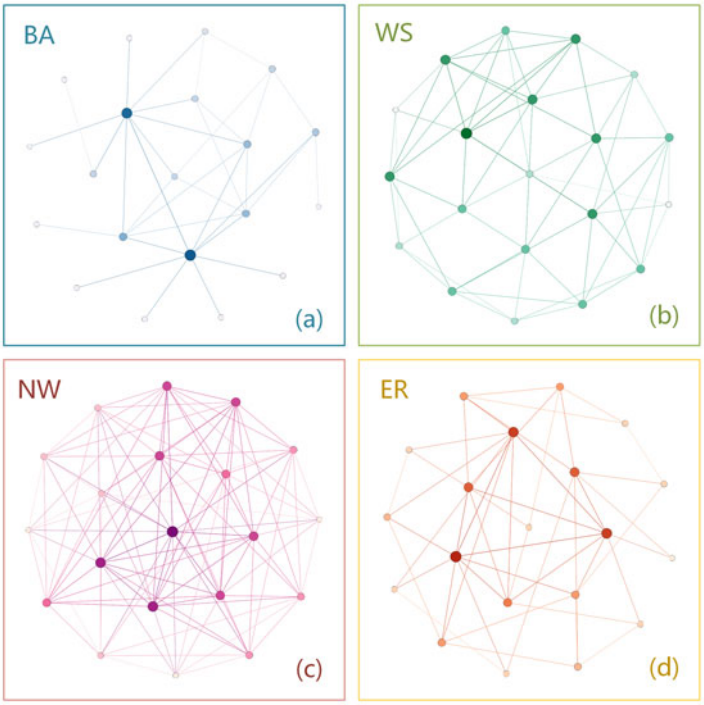


Fig. 2.3 An example of four typical network models. (**a**) BA scale-free network. (**b**) WS small-world network. (**c**) NW small-world network. (**d**) ER random network

- (Linear) Preferential Attachment: The above-referred incoming new node is simultaneously connected to each of the ne existing nodes, according to the following probability: for node u of degree k_u, $\prod_u = \frac{k_u}{\sum_{v=1}^{N_t} k_v}$, where N_t denotes the total number of current existing nodes. The process of preferential attachment is terminated until the total number of nodes reaches the target value.

A WS small-world network can be generated by the WS algorithm:

- Start from a ring-shaped network with N_0 nodes, in which each node is connected to its $2K$ neighbors, K nodes on each side, where $K > 0$ is a small integer.
- For every pair of connected nodes in the ring-shaped network, rewire the edge in such a way that the beginning end of the edge is kept but the other end is disconnected with probability p and then reconnected it to a node randomly chosen from the network.

The NW algorithm is as follows:

- Start from a ring-shaped network with N_0 nodes, in which each node is connected to its $2K$ neighbors, where $K > 0$ is an integer (usually small).

- For every pair of originally unconnected nodes, with probability p ($0 < p \ll 1$), add an edge to connect them.

An ER random network is generated as follows:

- Initialization: Start with N isolated nodes and N is the total number of nodes.
- Pick up all possible pairs of nodes, once and once only, from the N given nodes, and connect each pair of nodes by an edge with probability $p \in (0, 1)$.

The related parameters and their explanations are shown in the following.

- S_i: the adjacency matrix of infrastructure network i, $i = 1, 2, 3, 4$
- NI: the number of infrastructure networks, $NI \geq 2$
- S_{ij}: the adjacency matrix of interconnections between infrastructure network i and j, $i \neq j$
- b_n^i: the betweenness centrality of node n in infrastructure network i
- φ_n^i: the selection fitness of node n in infrastructure network i
- N_i: the number of nodes in infrastructure network i, $i = 1, 2, 3, 4$
- N_{ij}: the number of interconnections between infrastructure network i and j, $i \neq j$
- C_i: the set of connectors in infrastructure network i
- N_c^i: the number of connectors in infrastructure network i
- C_n^i: the connector n in C_i, $n = 1, 2, 3, 4, \ldots, N_c^i$
- o : the quota of interconnections for a connector
- e_{nl}: Binary variable. If an interconnection has been created between node n in infrastructure network i and node l in infrastructure network j, $e_{nl} = 1$; otherwise, $e_{nl} = 0$

2.1.2 On-Demand Cooperation Strategy

To describe the process of on-demand cooperation strategy in detail, the description of steps is given in Algorithm 1. It contains two stages of establishing the interconnection structure, selecting nodes which are qualified for connectors, and creating interconnections between the pairs of connectors. These two stages are realized based on selection fitness and two-sided matching, which can be adapted to different demands by changing variables.

Matching theory, also known as search and matching theory, provides mathematically tractable solutions for problems in economics [35]. It has been used to describe the formation of mutually beneficial relationships, such as labor relations and other human relationships like marriage [16, 24]. Recently, matching theory has been used to solve the basic wireless resource management problem [26]. The concept called two-sided matching is used to optimally match resources and users given their individual objectives and learned information [17].

In the game-theoretic analysis of two-sided matching markets, the phrase two-sided refers to two disjoint sets of agents, e.g., firms or workers. The term matching

refers to the bilateral nature of exchange in these markets, e.g., the worker works for some firms, then that firm employs the worker. In the process of solving the two-sided matching problem between firms and workers, each firm and each worker starts by building its preference list based on some necessary information of the other set. For example, firms build their preferences over workers based on the skills and experience of workers. After setting up the preferences, proper matching algorithms must be developed to achieve the required system objectives such as maximizing the satisfaction degree between firms and workers. Since two-sided matching can overcome some limitations of game theory and optimization, we use it to describe the formation of interconnections between two different infrastructure networks.

In the method of selecting connectors, the selection fitness is modeled as a power law function of the betweenness centrality b based on the consideration that relative importance of node in the transmission efficiency is growing rapidly with the increase of betweenness centrality. Hence, the selection fitness φ_n^i for node n in the infrastructure network i is simply defined as

$$\varphi_n^i = \left(b_n^i\right)^{\alpha}. \tag{2.2}$$

The parameter α is a variable reflecting the extent of tendency for nodes with higher betweenness centrality to be selected as connectors, which has a great influence on the number of connectors. Equation (2.2) is used to preferentially select connectors for the creation of potential interconnections between the infrastructure network i and the others. A node which is more helpful to satisfy the demand has a higher φ_n^i and receives a higher chance of having an interconnection as a connector.

Algorithm 1 The on-demand cooperation algorithm

Require: $S_i, i = 1, 2, 3, 4$
Ensure: $S_{ij}, i \neq j$
1: Calculating the betweenness centrality b_n^i of each node according to S_i.
2: Calculating the selection fitness φ_n^i of each node in infrastructure network i using Eq. (2.2).
3: Selecting nodes with higher selection fitness φ_n^i as connectors using Algorithm 2.
4: With the number of interconnections N_{ij}, matching two sets of connectors (C_i and C_j) in infrastructure network i and j using two-sided matching.
5: Placing these interconnections between N_{ij} pairs of connectors.
6: Updating the values of elements of S_{ij}.

Algorithm 2 provides a method of choosing a set of nodes to create interconnections, which should be beneficial for meeting the demand of slices. This method is based on the node-based structural characteristics of the infrastructure networks. Corresponding to specific demand of the slices, the method of nodes selection provides suitable selection criteria to select relatively optimal nodes as connectors to have interconnections. Then, the optimal location of creating interconnections need to be determined for each pair of infrastructure networks. Selecting the optimal pair

of connectors for each interconnection is achieved with two-sided matching theory which provides a decision method to match connectors.

A joint venture InP is the result of cooperation among NI infrastructure networks. In this paper, multiple infrastructure networks are represented by four network models for analyzing the impact of different topologies on cooperation performance. Hence, $NI = 4$ in our simulations and the interconnections are created among four networks. It should be noted that Algorithm 1 here is suitable for any values of NI, and the analysis of the simulations could be used as an efficient benchmark or reference.

For these four network models S_1, S_2, S_3, S_4, S_1 is a BA network, S_2 is a WS network, S_3 is a NW network, S_4 is a ER network. After calculating the selection fitness for each node in each infrastructure network, the selection of connectors is implemented by preferential picking. The algorithm for the implementation of preferential picking is introduced in Algorithm 2.

Algorithm 2 The implementation algorithm of preferential picking

Require: φ_n^i
Ensure: C_i
1: Define $\varphi_o^i = \sum_n \varphi_n^i$ for the infrastructure network i.
2: Define an interval I_n^i for node n and place them end to end in sequence.

$$I_n^i = \left(\sum_{l=1}^{n-1} \varphi_l^i, \sum_{l=1}^{n} \varphi_l^i \right)$$

and $\left\| I_n^i \right\| = \varphi_n^i$.
3: Construct the interval $\left(0, \varphi_o^i\right)$ from N_i smaller, non-overlapping, continuous intervals.

$$\left(0, \varphi_o^i\right) = \bigcup_{n=1}^{N_i} I_n^i$$

.
4: Generate random number r^i in the interval $\left(0, \varphi_o^i\right)$
5: Identify the component intervals I_n^i in which r^i lie.
6: Repeat step 4-5 for $\frac{N_i}{10}$ times. Hence, $N_c^i \leq \frac{N_i}{10}$.

We would like to caution here that the forms of all functions in the our method are an overly simple choice which is helpful to the final results, and different demands of slices might motivate a different forms of Eq. (2.2). For example, φ_n^i might not only depend on betweenness centrality, but also other structural characteristics such as degree and clustering coefficient. Furthermore, Eq. (2.2) would contain two characteristics or even more and not just single characteristic as in current example. For the slices which require high computing resources and low congestion rate (e.g., massive machine-type communications), the selection fitness needs to consider both the capacity and the degree.

After selecting the sets of connectors, the next step is to determine which pairs of connectors are suitable to create interconnections. Before creating interconnections between a pair of infrastructure networks, the number of interconnections should

be a exact value. The number of interconnections, as a quantity that reflects the degree of cooperation, is influenced by many practical factors such as costs and facility performance. Hence, the number of interconnections between each pair of infrastructure networks is predetermined in our simulations.

Given the number of interconnections, deciding the location of interconnections can be posed as a matching problem between pairs of connectors. The archetypal matching problem involving preferences is first introduced by David Gale and Lloyd Shapley in 1962 [16]. A matching in our model is essentially creating an interconnection between a pair of connectors. The main goal of matching is to optimally match connector pairs given their individual, often different, objectives and structural information. Depending on the scenario, each connector has a quota that defines the maximum number of connectors with which it can be matched. Each connector from infrastructure network i builds a ranking of the connectors from infrastructure network j using a preference relation.

The concept of a preference represents the individual view of each connector in the other set of connectors, based on their node-based structural characteristics. In its basic form, a preference can simply be defined in terms of structural characteristics which are beneficial to meet the demand. However, a preference is more generic than structural characteristics in that it can incorporate additional qualitative measures extracted from the information available to connectors according to the features of particular actual scenarios. For example, the processing and storage capacity of connectors need to be observed if the demand of slice is low mobility and higher user data rate (e.g., enhanced mobile broadband). The basic solution concept for a matching problem is the so-called two-sided matching [21]. In order to solve two-sided matching problems with preference information, the satisfaction degree m_{nl}^{i} of connector n in infrastructure network i with connector l in infrastructure network j is defined as preference based on the betweenness centrality b_l^j.

Moreover, the satisfactions functions for each pair of infrastructure networks is defined based on the demand of slices and the effect of betweenness centrality on satisfactions is analyzed by means of a linear function. In order to make the relatively importance of connectors from different infrastructure networks comparable, they are placed on the same scale using a min-max transform. The satisfaction degree m_{nl}^{i} is defined as

$$m_{nl}^{i} = \gamma_1 \left[\frac{b_l^j - \min\limits_l \left(b_l^j \right)}{\max\limits_l \left(b_l^j \right) - \min\limits_l \left(b_l^j \right)} \right]. \tag{2.3}$$

Similarly, we define the satisfaction degree w_{nl}^{j} of connector l in infrastructure network j with connector n in infrastructure network i based on the b_n^i.

$$w_{nl}^{j} = \gamma_2 \left[\frac{b_n^i - \min\limits_n \left(b_n^i \right)}{\max\limits_n \left(b_n^i \right) - \min\limits_n \left(b_n^i \right)} \right] \tag{2.4}$$

In both Eqs. (2.3) and (2.4), γ_1 and γ_2 are variables reflecting the extent of tendency for connectors with higher betweenness centrality to have a higher satisfaction degree. γ_1 denotes the impact extent of normalized betweenness centrality on the satisfaction degree m_{nl}^i, and γ_2 has the same meaning to w_{nl}^j. With the satisfaction degree m_{nl}^i and w_{nl}^j, the matrix of preference between infrastructure network i and infrastructure network j can be denoted by

$$P_{ij} = \left[\left(m_{nl}^i, w_{nl}^j\right)\right]_{N_c^i \times N_c^j}. \tag{2.5}$$

Actually, the functional forms of satisfaction degree m_{nl}^i and w_{nl}^j are not necessarily the same. Noticed that Eqs. (2.3) and (2.4) give a higher chance to place interconnections between connectors n and l whose values of betweenness centrality are higher and similar, but in general it is possible to use the other forms and variable parameters considering the diversity of actual situations as well, and the normalization function also can be replaced by the unit standards deviation, normalization to mean zero and others.

Considering that the creation of interconnections should improve the transmission efficiency, the aims of this two-sided matching are maximizing the sum of satisfactions and minimizing the differences of satisfactions for connectors. The solution of this two-sided matching problem is establishing a multi-objective optimization model. There are many techniques to deal with multiple objectives [23, 37, 40], in which the linear weighting method has been widely used [29]. For solving the model easily, the multi-objective model is normalized and transformed into single objective model due that the objectives are different in their scales.

Therefore, the multi-objective optimization model is built as follows.

$$\begin{aligned}
&\max f_1 = \sum_{n=1}^{N_c^i} \sum_{l=1}^{N_c^j} m_{nl}^i \cdot e_{nl} \\
&\max f_2 = \sum_{n=1}^{N_c^i} \sum_{l=1}^{N_c^j} w_{nl}^j \cdot e_{nl} \\
&\min f_3 = \sum_{i=1}^{N_c^i} \sum_{j=1}^{N_c^j} \left| m_{nl}^i - w_{nl}^j \right| \cdot e_{nl} \\
&s.t. \begin{cases} \sum_{n=1}^{N_c^i} e_{nl} \le o, \forall n \\ \sum_{l=1}^{N_c^j} e_{nl} \le o, \forall l \\ \sum_{n=1}^{N_c^i} \sum_{l=1}^{N_c^j} e_{nl} = N_{ij}, \forall n, l \\ e_{nl} = 0, 1 \end{cases}
\end{aligned} \tag{2.6}$$

In the above model, there are three objectives. The first objective function is to maximize the satisfaction degree of connectors in the infrastructure network i. The second objective function is to maximize the satisfaction degree of connectors in the infrastructure network j. That is, if the satisfaction degrees of a connector C_n^i in the infrastructure network i and a connector C_l^j in the infrastructure network j are very high, the possibility of matching C_n^i and C_l^j is high. The third objective function is to minimize the difference of the satisfaction degree between those two connectors.

Aside from the objectives, the first constraint is to guarantee that each connector in the infrastructure network i creates interconnections with o connectors at most. In our simulations, the value of quota o is 2, which means that the number of interconnections belonging to a connector is under 2. Similarly, the second constraint is to guarantee the quota of each connector in the infrastructure network j. The third constraint is to guarantee that the total number of interconnections is equal to the preset value N_{ij}. In the last constraint, $e_{nl} = 0$ represents that C_n^i and C_l^j are not matched, and $e_{nl} = 1$ represents that they are matched, which means that an interconnection has been created between them.

There are many techniques to deal with multiple objectives, in which the linear weighting method has been widely used. Multiple objectives are combined to single objective by the sum of weighted objectives, and we have

$$\max F = \frac{1}{2}(f_1 + f_2) - f_3. \tag{2.7}$$

Considering that the three objectives in our model are different in their scales, we normalize $\frac{1}{2}(f_1 + f_2)$ and f_3 by defining $R_{xy} = (r_{nl})_{N_c^i \times N_c^j}$ and $T_{xy} = (t_{nl})_{N_c^i \times N_c^j}$.

$$\begin{aligned} r_{nl} &= \frac{m_{nl}^i + w_{nl}^j}{2} \\ t_{nl} &= \left| m_{nl}^i - w_{nl}^j \right| \end{aligned} \tag{2.8}$$

and normalizing R_{xy} and T_{xy},

$$\begin{aligned} r'_{nl} &= \frac{r_{nl} - \min\left\{(r_{nl})_{N_c^i \times N_c^j}\right\}}{\max\left\{(r_{nl})_{N_c^i \times N_c^j}\right\} - \min\left\{(r_{nl})_{N_c^i \times N_c^j}\right\}} \\ t'_{nl} &= \frac{t_{nl} - \min\left\{(t_{nl})_{N_c^i \times N_c^j}\right\}}{\max\left\{(t_{nl})_{N_c^i \times N_c^j}\right\} - \min\left\{(t_{nl})_{N_c^i \times N_c^j}\right\}} \end{aligned}. \tag{2.9}$$

Thus, we have the normalized matrices

$$R'_{xy} = \left(r'_{nl}\right)_{N_c^i \times N_c^j}$$
$$T'_{xy} = \left(t'_{nl}\right)_{N_c^i \times N_c^j} \tag{2.10}$$

with which the combined matrix of satisfaction degree $C'_{xy} = \left(c'_{nl}\right)_{N_c^i \times N_c^j}$, $c'_{nl} = r'_{nl} - t'_{nl}$. Finally, the multi-objective optimization model is transformed into the single objective

$$\max F' = \sum\nolimits_{n=1}^{N_c^i} \sum\nolimits_{l=1}^{N_c^j} c'_{nl} \cdot e_{nl} \tag{2.11}$$

and the traditional linear programming method is adopted to solve this model.

2.2 Virtualization of Multi-Domain Infrastructures

2.2.1 Multi-Layer Complex Network Model

The rapid development of Internet and communication technology brings people into the "network era." We can see all kinds of networks everywhere in our life, such as transportation network, power network, trade network, and so on. The network has penetrated into the surrounding of human beings. As an individual, each person is a component unit of various social network relations, and as a biological system, it is also the result of biochemical reaction network. Networks introduced in network science can be objects in Euclidean space, such as the Internet, highway or subway systems, and neural networks, or they can be entities defined in abstract space, such as networks of acquaintances and networks of collaboration among scientists. These networks gradually evolve into complex systems with complex relationships with the evolution process and scale expansion. The research on complex systems has attracted great interest of scholars. Initially, the study of networks was mainly a branch of discrete mathematics, namely graph theory. The upsurge of complex network research focused on the decade at the end of the twentieth century. The research on the irregularity, complexity and dynamic evolution behavior of network structure began to rise, and the focus of the research shifted from the analysis of small network structure to the large network system composed of thousands of nodes. At present, there are a large number of review articles [12, 22] and books on complex networks [8, 30] for reference.

Complex network, as a high abstraction of a large number of real complex systems, has played a great role in promoting the research of real networks. In the research of complex network science, individuals in complex systems are regarded as nodes in complex networks, and the dependencies and interactions between individuals are regarded as connections between nodes. Based on this, a complex network which can abstractly represent the complex system is established

to solve the specific key problems in the actual complex system from the perspective of network structure. From the social relationship between people [6], scientific collaboration networks [33], airport network [20], biological network [25], to the large-scale Internet and World Wide Web [1], complex network theory provides a large number of mathematical models and analytical conclusions [38] for the study of the structural characteristics of real network systems [10] and the analysis of dynamic behavior problems [4]. Nowadays, the analysis method of complex network has been applied in various fields to describe the individual characteristics and the overall behavior of the actual complex system.

The traditional method in complex network research is mainly a simple representation of the actual complex system. The components or basic units of each system are abstracted into a network node, and the interaction between each unit pair can be regarded as a connection with certain weight. However, because the interaction relationship between unit pairs in the actual system does not belong to the same type, simply using the connection of the same status to represent this relationship will cause the information in the system cannot be fully captured and even lead to the wrong judgment of the real problems. Therefore, with the in-depth study of the characteristics and behavior of traditional complex networks, researchers in the area of network science have been committed to study multilayer complex networks. Each layer of multilayer complex networks is abstracted with different behavior and attributes, which is contribute to solve the complex connection relationship between different entities in the single layer and make the relationship in the network more clearly visible. Therefore, it is widely used in various fields, such as transportation, power, computer, biology, and other fields. Compared with single-layer complex network, multilayer complex network can more accurately grasp the complexity of the actual network and analyze the dynamic behavior of different types of connections in the actual multilayer system.

A simple model of multilayer complex network is shown in Fig. 2.4. According to the definition given in reference [5], a multilayer network can be denoted as $\mathcal{M} = (\mathcal{G}, \mathcal{C})$ where $\mathcal{G}$ is a family of (directed or undirected, weighted or unweighted) graphs $G_\alpha = (X_\alpha, E_\alpha)$ called layers of $\mathcal{M}$ and

$$\mathcal{C} = \left\{ E_{\alpha\beta} \subseteq X_\alpha \times X_\beta; \alpha, \beta \in \{1, \cdots, M\}, \alpha \neq \beta \right\} \tag{2.12}$$

is the set of interconnections between nodes of different layers G_α and G_β. The elements of $\mathcal{C}$ are called *crossed layers*, and the elements of each E_α are called *intralayer* connections of $\mathcal{M}$ in contrast with the elements of each $E_{\alpha\beta}$ $(\alpha \neq \beta)$ that are called *interlayer* connections. A multilayer network model is schematically illustrated in Fig. 2.4. The colored layers G_1, G_2, and G_3 are the different subsets of the underlying layer G_0. The solid lines represent the intralayer links and the dotted lines represent the interlayer links (interconnections). It is noted that the interlayer links between the underlying layer and upper layers are not displayed entirely, and the interlayer links between the upper layers are also not displayed in this figure.

Based on the universal model definition of multilayer complex networks, the description of the topological properties of multilayer complex networks has

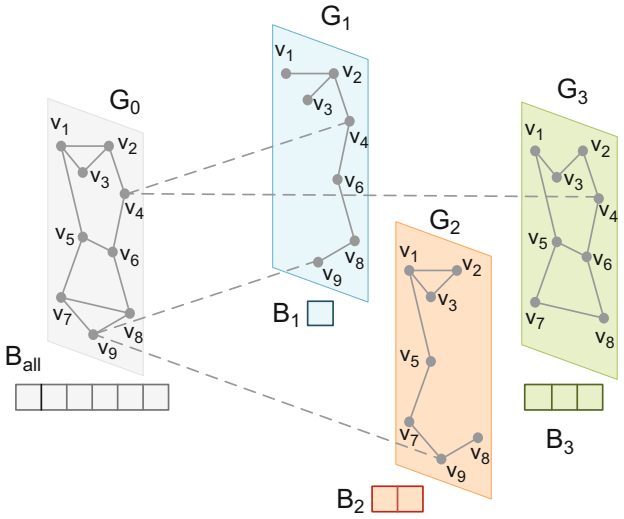


Fig. 2.4 The simple model of multilayer complex network

evolved from the simple weighted combination of single-layer complex networks to the establishment of a unique description scheme of topological properties of multilayer complex networks. The solid research foundation and sufficient methods of multilayer complex network theory have important contributions to obtaining the structural property and analyzing the collaboration among slices. The realization of network slicing can be abstracted as the generation of a multilayer network with nodes and edges organized into multiple layers, where each layer can be described as a set of VNFs with some pattern of interconnections between them.

2.2.2 Mathematical Description of Infrastructure Networks

In the scenario of using a multilayer network model to describe network slicing, the mapping relationship between the underlying infrastructure network and slices can be denoted by interconnections between the underlying layer and the upper layers. The structure of interconnections determines the deployment of VNFs and depicts the resource competition among them, which influence the traffic performance of slices. The infrastructure network can be represented by the underlying layer G_0, while slices can be represented by $G_m, m = 1, 2, 3, \cdots$. The nodes of the underlying layer G_0 denote the physical standardized servers of the infrastructure network and the nodes of layers G_m denote the VNFs of slices. Here only focuses the relationship between the infrastructure network and multiple slices, thus there are no interconnections between different slices, which is different from other multilayer network model.

Hence, in order to distinguish the underlying layer from the upper layers, $\mathcal{NS} = (\mathcal{P}, \mathcal{V}, \mathcal{L})$ is used to denote the slices. $\mathcal{P}$ denotes the infrastructure network, $\mathcal{V}$ denotes slices, and $\mathcal{L}$ denotes the interconnections between the infrastructure network and slices. These interconnections reflect the mapping of VNFs in slices to the underlying physical servers. For network slicing in the single infrastructure network which is similar as the multilayer network in Fig. 2.4, $\mathcal{P} = G_0$ and $\mathcal{V}=\{G_m, m = 1, 2, 3, \cdots .\}$. Note that node V_9 in G_1 and its counterpart in G_2 are competing the resources of node V_9 in G. The total resources B_{all} of layer G_0 can be divided into three parts B_1, B_2, and B_3 and be allocated to three layers, respectively.

For the multi-tenant slicing in the joint infrastructure network which are formed by multiple different infrastructure networks, the representation of slicing can be denoted as $\mathcal{NS} = (\mathcal{P}, \mathcal{C}, \mathcal{V}, \mathcal{L})$ where $\mathcal{C}$ represents the cooperation links between each pair of infrastructure networks. In this general expression, assuming that there are N_P InPs jointly providing physical resources through resource sharing in order to reduce CAPEX/OPEX, each InP owns the infrastructure network P_β, thus we have

$$\mathcal{P} = \left\{P_\beta; \beta \in \{1, \cdots, N_P\}\right\}, \tag{2.13}$$

where $P_\beta = \left(PX_\beta, PE_\beta\right)$. Considering that the physical nodes of the infrastructure network include the BSs of RAN, optical switches (OSs) of transport network and the core nodes (CNs) of core network, the physical nodes can be represented by

$$PX_\beta = PX_\beta^{BS} \cup PX_\beta^{OS} \cup PX_\beta^{CN}. \tag{2.14}$$

The cooperation links between P_β and P'_β can be denoted as

$$\mathcal{C} = \left\{E_{\beta\beta'} \subseteq PX_\beta \times PX_{\beta'};\ \beta, \beta' \in \{1, \cdots, N_P\}, \beta \neq \beta'\right\}. \tag{2.15}$$

For slices deployed in the joint infrastructure network,

$$\mathcal{V} = \{V_\alpha; \alpha \in \{1, \cdots N_V\}\}, \tag{2.16}$$

where $V_\alpha = (VX_\alpha, VE_\alpha)$. Similar to the infrastructure network, slices are also composed of three domains, thus

$$VX_\alpha = VX_\alpha^{BS} \cup VX_\alpha^{OS} \cup VX_\alpha^{CN}. \tag{2.17}$$

Hence, the interconnections between N_V slices and N_P infrastructure networks can be expressed by

$$\mathcal{L}=\left\{E_{\beta\alpha} \subseteq PX_\beta \times VX_\alpha;\ \beta \in \{1, \cdots, N_P\}, \alpha \in \{1, \cdots, N_V\}\right\}. \tag{2.18}$$

2.3 Federated Management Framework of Physical Resources

2.3.1 Multi-Domain Slice Management Model

Figure 2.5 shows the management model of multi-domain slices. A infrastructure network resource domain represents an administrative domain in a network belonging to a MNO/InP. The infrastructure resources are hardware and software components that can be found in a data center or a mobile network of the MNO/InP. There are some functions in this model that can be classified to network slice design components or multi-domain orchestrator components. Network slice design studio and network slice ordering in the network slice manager, and descriptor design function belong to network slice design components. Network slice life-cycle management function, multi-domain deployment executor, network slice operations belong to multi-domain orchestrator components. In addition, there are two supporting functions: the network slice policy function (NPF) for policies management and the network slice database (NSLDB) for storing all information that is created and used by other functional blocks. The process engine manages and monitors the state of activities from order, design, and creation to termination of NSIs.

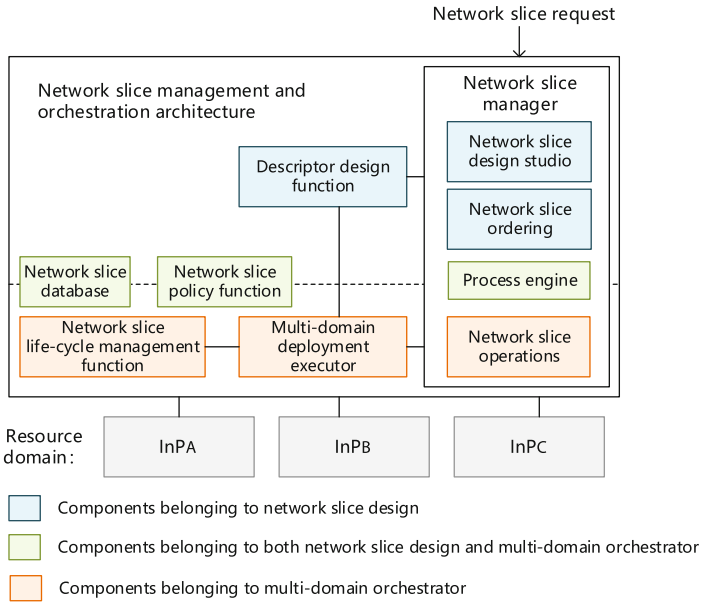


Fig. 2.5 The management model of multi-domain slices

The input of the network slice design studio is the network slice request, and the output is the network slicing descriptor. Network slice design studio provides tools and modeling primitives to a user for correctly defining and on-boarding the information needed to create a new network slice type. Network slice ordering provides users with templates for each network slice type. Network slice operations enable operators to monitor and interact with network slices. The network slice manager forward the network slice requests to the descriptor design function. The descriptor design function is able to build a NSI based on the network slice requests, create a customer-specific network slice descriptor, and verify it. The network slice manager controls the process of network slice design and orders a deployment of the NSI. The descriptor design function on-boards this network slice descriptor to the multi-domain deployment executor. The network slice descriptor contains all information needed to deploy, configure, activate, and operate a NSI during its life cycle. The multi-domain deployment executor extracts the resource domain descriptors, network function configuration data and workflows from the network slice descriptor and then sends the resource domain descriptors to the appropriate resource managers to deploy network functions and other resources. After successful deployment, configuration, and activation of NSI, the network slice manager gives the control over network slice lifecycle to network slice life-cycle management function which is responsible to resolve resource allocation conflicts between network slices.

2.3.2 Federated Infrastructure Management Framework

Based on the management model of multi-domain slices, the federated management framework of multi-domain infrastructures is shown in Fig. 2.6. The main fundamental components and functions include service broker plane [36], multi-domain service conductor plane, network slice orchestration plane, and multi-domain infrastructures.

The service broker is responsible for handling incoming slice requests from tenants (e.g., verticals, MVNOs, and application providers). It collects business, policy, and administrative information by interacting with the OSS/BSS. A global service support repository is created with the abstracted service capability information regarding different administrative domains collected by service broker. When a slice request arrives, service broker performs the admission control and negotiation with the requesting tenant considering the OSS/BSS policy and rules.

Successful requests are forwarded to the multi-domain service conductor plane which is responsible for service orchestration and management across federated resources. It consists of two main building blocks, service conductor and cross-domain slice coordinator. Service conductor decomposes the successful slice request toward different administrative domains and decides on the combination of domains according to the service requirements. It also instantiates a cross-domain slice coordinator for the newly allocated multi-domain NSI. The cross-domain slice

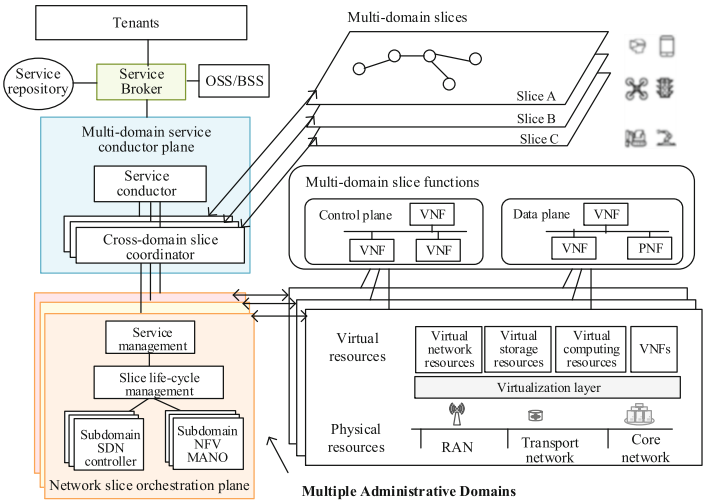


Fig. 2.6 The federated management framework of multi-domain infrastructures

coordinator monitors, manages, and controls the corresponding resources related with the federated NSI.

The network slice orchestration plane interacts with the cross-domain slice coordinator. It allocates resources in the relative domains for the federated NSI and provides corresponding lifecycle management through the functional blocks including service management function, slice lifecycle management function, sub-domain NFV MANO, and sub-domain SDN controller. The service management function analyzes the slice request received from the cross-domain slice coordinator and feeds back the service and performance capability information related with the underlying resources. In addition, the service management function identifies the RAN and core network functions and determines logical links characterized by bandwidth, delay, jitter, packet loss, and so on. The slice lifecycle management function identifies the appropriate network slice template and forms a logical network graph mapped in the underlying infrastructures. It is also responsible for the instantiation, run-time, and orchestration of a network slice subnet instance (NSSI). The sub-domain NFV MANO provides an abstracted view of the underlying infrastructure to the slice lifecycle management function and performs the instantiation and run-time operations of the corresponding VNF, computation, or storage slates. The sub-domain SDN controller provides the network connectivity and service chaining among the allocated VNFs. It also feeds the slice lifecycle management function with an abstracted network resource view and monitoring reports for assuring the desired service level agreement (SLA).

In the multi-domain infrastructures, the physical resources consist of processing, storage and network resources in the RAN domain, transport domain, and core network domain. The virtualization layer is responsible for abstracting the

underlying hardware resources and decoupling VNFs from hardware. The virtual resources are the abstracted physical resources that VNFs are running directly on. These VNFs include control plane and data plane functions are deployed in the hardware of the infrastructures. With the federated management framework of multi-domain infrastructure, the operational procedures of multi-domain network slice configuration and modification are elaborated in [39].

Acknowledgments If you want to include acknowledgments of assistance and the like at the end of an individual chapter please use the `acknowledgement` environment—it will automatically render Springer's preferred layout.

References

1. Albert, R., Jeong, H., Barabási, A.L.: Internet: Diameter of the world-wide web. Nature **401**(6), 130–131 (1999)
2. Antonopoulos, A., Kartsakli, E., Bousia, A., Alonso, L., Verikoukis, C.: Energy-efficient infrastructure sharing in multi-operator mobile networks. IEEE Commun. Mag. **53**(5), 242–249 (2015)
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)
4. Barrat, A., Barthelemy, M., Vespignani, A.: Dynamical processes on complex networks[M]. Cambridge University Press, Cambridge (2008)
5. Boccaletti, S., Bianconi, G., Criado, R., Del Genio, C.I., Gómez-Gardeñes, J., Romance, M., Sendiña-Nadal, I., Wang, Z., Zanin, M.: The structure and dynamics of multilayer networks. Phys. Rep. **544**(1), 1–122 (2014)
6. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: Structure and dynamics. Phys. Rep. **424**(4), 175–308 (2006). https://doi.org/10.1016/j.physrep.2005.10.009. http://www.sciencedirect.com/science/article/pii/S037015730500462X
7. Bollobás, B.: Random Graphs. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge (2001)
8. Bornholdt, S., Schuster, H.G.: Handbook of graphs and networks[J]. From Genome to the Internet. Willey-VCH, Weinheim (2001). https://onlinelibrary.wiley.com/doi/book/10.1002/3527602755
9. Cano, L., Capone, A., Carello, G., Cesana, M., Passacantando, M.: On optimal infrastructure sharing strategies in mobile radio networks. IEEE Trans. Wireless Commun. **16**(5), 3003–3016 (2017)
10. Cohen, R., Havlin, S.: Complex networks: structure, robustness and function[M]. Cambridge University Press, Cambridge (2010)
11. Devlic, A., Hamidian, A., Liang, D., Eriksson, M., Consoli, A., Lundstedt, J.: Nesmo: Network slicing management and orchestration framework. In: 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1202–1208 (2017). https://doi.org/10.1109/ICCW.2017.7962822
12. Dorogovtsev, S.N., Mendes, J.F.F.: Evolution of networks. Adv. Phys. **51**(4), 1079–1187 (2001)
13. Erd6s, P., Rényi, A.: On the evolution of random graphs. Publ. Math. Inst. Hungar. Acad. Sci **5**, 17–61 (1960)
14. Foukas, X., Nikaein, N., Kassem, M.M., Marina, M.K., Kontovasilis, K.P.: Flexran: A flexible and programmable platform for software-defined radio access networks. In: CoNEXT, pp. 427–441 (2016)
15. Foukas, X., Patounas, G., Elmokashfi, A., Marina, M.K.: Network slicing in 5G: Survey and challenges. IEEE Commun. Mag. **55**(5), 94–100 (2017). https://doi.org/10.1109/MCOM.2017.1600951

16. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. Am. Math. Monthly **69**(1), 9–15 (1962)
17. Gu, Y., Saad, W., Bennis, M., Debbah, M., Han, Z.: Matching theory for future wireless networks: fundamentals and applications. IEEE Commun. Mag. **53**(5), 52–59 (2014). Preprint (2014). arXiv:1410.6513
18. Guan, W., Wen, X., Wang, L., Lu, Z.: Network slicing management of 5G ultra-dense networks based on complex network theory. In: 2017 IEEE Globecom Workshops (GC Wkshps), pp. 1–6 (2017). https://doi.org/10.1109/GLOCOMW.2017.8269197
19. Guan, W., Wen, X., Wang, L., Lu, Z.: On-demand cooperation among multiple infrastructure networks for multi-tenant slicing: a complex network perspective. IEEE Access **6**, 78689–78699 (2018). https://doi.org/10.1109/ACCESS.2018.2885143
20. Guimerá, R., Amaral, L.A.N.: Modeling the world-wide airport network. Eur. Phys. J. B **38**(2), 381–385 (2004)
21. Gusfield, D., Irving, R.W.: The Stable Marriage Problem: Structure and Algorithms. MIT Press, Cambridge, MA (1989)
22. Heidelberg, S.V.B.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**(1), xii (2001)
23. Hwang, C.L., Yoon, K.: Lecture Notes in Economics and Mathematical Systems: Multiple Attribute Decision Making: Methods and Appllication. Springer, New York (1981)
24. Irving, R.W., Leather, P., Gusfield, D.: An efficient algorithm for the "optimal" stable marriage. J. ACM (JACM) **34**(3), 532–543 (1987)
25. Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., Barabási, A.L.: The large-scale organization of metabolic networks. Nature **407**(6804), 651–654 (2000)
26. Jorswieck, E.A.: Stable matchings for resource allocation in wireless networks. In: Digital Signal Processing (DSP), 2011 17th International Conference on, pp. 1–8. IEEE (2011)
27. Kim, Y-b., Hong, B., Choi, W.: Scale-free wireless networks with limited degree information. IEEE Wireless Commun. Lett. **1**(5), 428–431 (2012). https://doi.org/10.1109/WCL.2012.061212.120309
28. Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. J. Complex Networks **2**(3), 203–271 (2014)
29. Li, Y.H., Fan, Z.P., Chen, X., Kang, F.: A multi-objective optimization model for matching ventures and venture capitalists. In: 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4. IEEE (2008)
30. Dorogovtsev, S.N., Mendes, J.F.F.: Evolution of networks: From biological nets to the Internet and WWW[M]. Oxford University Press (2003)
31. Newman, M.E.: The structure and function of complex networks. SIAM Rev. **45**(2), 167–256 (2003)
32. Newman, M.E., Watts, D.J.: Renormalization group analysis of the small-world network model. Phys. Lett. A **263**(4), 341–346 (1999)
33. Newman, M.E.J.: Scientific collaboration networks. I. network construction and fundamental results. Phys. Rev. E **64**, 016131 (2001). https://doi.org/10.1103/PhysRevE.64.016131. https://link.aps.org/doi/10.1103/PhysRevE.64.016131
34. Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J.J., Lorca, J., Folgueira, J.: Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges. IEEE Commun. Mag. **55**(5), 80–87 (2017)
35. Roth, A.E., Sotomayor, M.: Two-sided matching. Handbook of Game Theory with Economic Applications, vol. 1, pp. 485–541. Cambridge University Press, Cambridge (1992)
36. Samdanis, K., Costa-Perez, X., Sciancalepore, V.: From network sharing to multi-tenancy: The 5G network slice broker. IEEE Commun. Mag. **54**(7), 32–39 (2016)
37. Steuer, R.E.: Manual for the Adbase Multiple Objective Linear Programming Package. University of Georgia, Athens (1992)
38. Strogatz, S.H.: Exploring complex networks. Nature **410**(8), 268–276 (2001)
39. Taleb, T., Afolabi, I., Samdanis, K., Yousaf, F.Z.: On multi-domain network slicing orchestration architecture and federated resource control. IEEE Network **33**(5), 242–252 (2019)

40. Vassilev, V., Narula, S.C.: A reference direction algorithm for solving multiple objective integer linear programming problems. J. Oper. Res. Soc. **44**(12), 1201–1209 (1993)
41. Vincenzi, M., Antonopoulos, A., Kartsakli, E., Vardakas, J., Alonso, L., Verikoukis, C.: Multi-tenant slicing for spectrum management on the road to 5G. IEEE Wireless Commun. **24**(5), 118–125 (2017). https://doi.org/10.1109/MWC.2017.1700138
42. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1998)
43. Zhang, B., Liu, R., Massey, D., Zhang, L.: Collecting the internet as-level topology. ACM SIGCOMM Comput. Commun. Rev. **35**(1), 53–61 (2005)
44. Zhang, S., Zhang, N., Zhou, S., Gong, J., Niu, Z., Shen, X.: Energy-aware traffic offloading for green heterogeneous networks. IEEE J. Sel. Areas Commun. **34**(5), 1116–1129 (2016)

Chapter 3
Intelligent Deployment and Orchestration of E2E Slices

3.1 Service-Oriented Slice Deployment Policy

Owing that different use case families have different demands on the multi-domain resources, the deployment of E2E slices should be service-oriented. In order to support a diverse set of use cases, the heterogeneous resources of multiple infrastructure networks need to be allocated dynamically. Besides, E2E slices need to be instantiated rapidly and should support cross-domain deployment. The deployment of E2E slices in essence is the allocation of virtual resources and the placement of VNFs. There are plentiful related researches on VNF placement, such as the algorithms proposed in [2, 11]. However, these algorithms are suitable for a single type of VNF chain request without considering diverse service requirements. In addition to the VNF placement problem, the virtual network embedding (VNE) problem [7] which focuses on the mapping from slices to the infrastructure networks has also been studied maturely in recent years. Nonetheless, little works in existing literature have been done on the deployment of E2E network slices. Therefore, we propose a service-oriented deployment policy of E2E network slices [5], improving the resource utilization by analyzing different features of use cases and the topological properties of infrastructure.

3.1.1 The Deployment Model of E2E Slices

In this section, we describe the mathematical model of network slicing including infrastructure network model, network slice request (NSR) model, and slice deployment model.

The deployment of NS requires topological information of infrastructure network including the structural characteristics of physical nodes, e.g., base stations (BS), optical switches (OS), core nodes (CON). The infrastructure network can

W. Guan, H. Zhang, *Network Slicing for Future Wireless Communication*, Wireless Networks, https://doi.org/10.1007/978-3-031-58229-5_3

Table 3.1 A summary of parameters for E2E slice deployment

Parameters	Definitions
$C_I^{CON}\left(n_I^{CON}\right)$	Computing resource of core node n_I^{CON}, $n_I^{CON} \in N_I^{CON}$;
$C_R^{CON}\left(n_R^{CON}\right)$	Computing resource requirement of the virtual core node $n_R^{CON} \in N_R^{CON}$;
$B_R^{wl}\left(e_R^{wl}\right)$	Bandwidth requirement of the virtual wireless link $e_R^{wl} \in E_R^{wl}$;
$B_R^{ol}\left(e_R^{ol}\right)$	Bandwidth requirement of the virtual optical link $e_R^{ol} \in E_R^{ol}$;

be abstracted as undirected weighted graph, which can be denoted as $G_I = (N_I, E_I, C_I, B_I)$. Similar to some previous literatures, we only take into consideration the capacity of nodes and bandwidth of links. N_I stands for the set of physical nodes, which can be partitioned into the set of base stations N_I^{BS}, the set of optical switches N_I^{OS}, and the set of core nodes N_I^{CON}, $N_I = N_I^{BS} \cup N_I^{OS} \cup N_I^{CON}$. E_I stands for the set of physical links including the wireless wave links E_I^{wl} and wired optical links E_I^{ol}, $E_I = E_I^{wl} \cup E_I^{ol}$.

$$E_I^{wl} = \bigcup_{n_I^{BS} \in N_I^{BS}} E_I^{wl}\left(n_I^{BS}\right), \tag{3.1}$$

where $E_I^{wl}\left(n_I^{BS}\right)$ is the subset of wireless link e_I^{wl} responsible of connecting n_I^{BS} and other nodes. C_I stands for the capacity of physical nodes, which includes BS wireless channel capacity C_I^{BS} and computing resource of core cloud C_I^{CON}, $C_I = C_I^{BS} \cup C_I^{CON}$. B_I stands for the bandwidth set of physical links, including the available bandwidth set of wireless wave links B_I^{wl} and wired optical links B_I^{ol}, $B_I = B_I^{wl} \cup B_I^{ol}$.

In our model, the set of NSRs consists of three types of slices for three use case families, which can be denoted by R_{NS}, $R_{NS} = R_e \cup R_m \cup R_u$. R_e represents eMBB slice, R_m represents mMTC slice, and R_u represents uRLLC slice. Each request is regarded as $G_R = (N_R, E_R, C_R, B_R, T_R)$ where N_R represents nodes of network slice, E_R represents links, C_R denotes capacity, B_R denotes bandwidth, and T_R is the duration of the NSR remaining in the infrastructure network. Thus, $G_{R_e} = \left(N_{R_e}, E_{R_e}, C_{R_e}, B_{R_e}, T_{R_e}\right)$ is for request R_e, and similarly G_{R_m} and G_{R_u} are for requests R_m and R_u, respectively.

Slice deployment is a process in which nodes of slice requests are mapped onto substrate nodes and links are mapped onto substrate paths on the premise of meeting service demands of slices. The mapping process consists of two stages, the node mapping and the link mapping. The node mapping represents the placement of VNFs while the link mapping is chaining those VNFs. A node of NSR can only be mapped on a node of infrastructure network, and a node of infrastructure network can only host a node from the same of NSR. Table 3.1 is a summary of parameters that are used for the formulation of the mathematical model and the introduction of the decision variables, and Table 3.2 is a summary of variables.

Table 3.2 A summary of variables for E2E slice deployment

Variables	Definitions
$C_I^{wl}\left(e_I^{wl}\right)$	Channel capacity of the wireless link e_I^{wl}, $e_I^{wl} \in E_I^{wl}$;
$B_I^{wl}\left(e_I^{wl}\right)$	Bandwidth assigned to the wireless link e_I^{wl};
$\mu_{n_R^{BS},n_I^{BS}}$	Binary variable, if n_R^{BS} of G_R is mapped to n_I^{BS} of G_I, $\mu_{n_R^{BS},n_I^{BS}} = 1$; Otherwise, $\mu_{n_R^{BS},n_I^{BS}} = 0$;
$\mu_{n_R^{OS},n_I^{OS}}$	Binary variable, if n_R^{OS} of G_R is mapped to n_I^{OS} of G_I, $\mu_{n_R^{OS},n_I^{OS}} = 1$; Otherwise, $\mu_{n_R^{OS},n_I^{OS}} = 0$;
$\mu_{n_R^{CON},n_I^{CON}}$	Binary variable, if n_R^{CON} of G_R is mapped to n_I^{CON} of G_I, $\mu_{n_R^{CON},n_I^{CON}} = 1$; Otherwise, $\mu_{n_R^{CON},n_I^{CON}} = 0$;
$\nu_{e_R^{wl},e_I^{wl}}$	Binary variable, if a virtual link e_I^{wl} of G_R traverse the physical wireless link e_I^{wl}, $\nu_{e_R^{wl},e_I^{wl}} = 1$; Otherwise, e_I^{wl}, $\nu_{e_R^{wl},e_I^{wl}} = 0$;
$\nu_{e_R^{ol},e_I^{ol}}$	Binary variable, if a virtual link e_I^{ol} of G_R traverse the physical optical link e_I^{ol}, $\nu_{e_R^{ol},e_I^{ol}} = 1$; Otherwise, e_I^{ol}, $\nu_{e_R^{ol},e_I^{ol}} = 0$;
$\xi_{n_R^{BS},e_I^{wl}}$	Binary variable, if n_R^{BS} of G_R is served by $e_I^{wl} \in E_I^{wl}\left(n_R^{BS}\right)$, $\xi_{n_R^{BS},e_I^{wl}} = 1$; Otherwise, $\xi_{n_R^{BS},e_I^{wl}} = 0$;

The deployment process of E2E slices consists of VNF placements and chaining VNFs. In order to achieve better resource efficiency and higher revenue of service provision, the topological properties of infrastructure networks are analyzed based on CN theory and combined in the VNF placement. The degree of a node measures the number of edges that connect to it, which reflects the level of influence. The node degree describes the number of its neighborhood nodes, which can be formulated as

$$d_i = \sum_{j \in N} \delta_{ij}. \tag{3.2}$$

The parameter δ_{ij} takes the value 1 if node i and node j are directly connected, otherwise it takes the value 0. The betweenness centrality quantifies how much a node is found between the path linking other pair of nodes. This measure can describe the importance of a node with respect to the shortest path. The betweenness centrality is defined as the fraction of shortest paths between any pair of nodes that travel through the node, which can be denoted by

$$b_i = \sum_{s \neq i \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}. \tag{3.3}$$

In this equation, σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(i)$ is the number of those paths that pass through node i.

Placing VNFs means to select the physical nodes of substrate network as host for the virtual nodes of NSRs under the condition of satisfying the capacity

requirements. According to the literatures of VNE problem, the local resources for nodes are measured by

$$NR(i) = C(i) \cdot \sum_{l \in s(i)} BW(l), \tag{3.4}$$

where $C(i)$ represents the capacity of node i, $s(i)$ represents the set of links that directly connected to the node i, $BW(l)$ represents the current available bandwidth of link l. The main shortcoming of this measurement is ignoring the topological characteristics of nodes. Hence, in the step of placing VNFs, we combine the degree and betweenness centrality of nodes to measure the importance of nodes in infrastructure network and slices.

First, the degree and betweenness centrality of nodes are normalized. Considering that the degree of node is not exceed $N-1$ when the total number of nodes is N, the normalization of the degree can be expressed by

$$d'_i = \frac{d_i}{N-1}. \tag{3.5}$$

Similarly, the betweenness centrality of node can be normalized by using

$$b'_i = \frac{2b_i}{(N-1)(N-2)} \tag{3.6}$$

because the maximum of $(N-1)(N-2)/2$. In the case of reaching maximum, each node pair of the network has at least one shortest path that travel through the node. Based on these normalized metrics of nodes, the weighting parameters of node i can be given by $\frac{d'_i + b'_i}{2}$. Therefore, combine the local resource and weighting of each node, the node importance of i can be given by

$$NI(i) = NR(i) \times \left(\frac{d'_i + b'_i}{2} \right). \tag{3.7}$$

According to the node importance, we use graphical breadth-first-search (BFS) algorithm to sort nodes and map the virtual nodes to physical nodes based on BFS. The sorting algorithm of virtual nodes is listed in Algorithm 3. Based on the sorting algorithm, the node mapping algorithm is introduced in Algorithm 4.

The procedure of creating paths that interconnect the VNFs placed nodes would be achieved on the basis of k-shortest paths (KSP) algorithm. KSP algorithm is used to select suitable physical paths in the premise of satisfying the bandwidth resource requirements. After removing the link paths that do not satisfy the requirements, Floyd algorithm is used to calculate the shortest path. More details are shown in Algorithm 5.

Algorithm 3 The sorting algorithm for virtual nodes in NSRs

Require: N_R: the set of virtual nodes in NSR
Ensure: N'_R: the sequence of sorted virtual nodes
1: Calculating NI value of each virtual node.
2: Sorting the virtual nodes by NI value in non-increasing order.
3: Selecting the virtual node with highest NI value as R.
4: Using R as the root node, traverse the graph of NSR using BFS algorithm, and get the BFS tree T.
5: Sorting the virtual nodes in each layer of T according to NI value in non-increasing order.
6: Return N'_R.

Algorithm 4 The node mapping algorithm based on BFS

Require: R_{NS}: the arrived NSR
Ensure: M_{node}: the results of node mapping
1: Sort virtual nodes with Algorithm 3.
2: Sort physical nodes according to their NI values in non-increasing order.
3: **for** each virtual node **do**
4: **if** it is root R **then**
5: it is mapped into the physical node with the greatest value of NI.
6: **else**
7: find the parent node P of it.
8: find the mapped physical node I for P.
9: find the neighbor nodes of I as the candidate physical nodes C.
10: choose one of C which owns the greatest value of NI in the premise of satisfying the capacity requirements.
11: **end if**
12: return M_{node}.
13: **end for**

Algorithm 5 The link mapping algorithm based on KSP

Require: R_{NS}: the arrived NSR
Ensure: M_{link}: the results of link mapping
1: Sort the virtual links according to bandwidth in non-increasing order.
2: **for** each virtual link l. **do**
3: calculate the bandwidth requirement $BW(l)$.
4: remove the physical links that can not meet the bandwidth requirement.
5: according to M_{node}, find the mapped physical nodes of l.
6: find the physical shortest path between these two physical nodes by using Floyd algorithm.
7: return M_{link}.
8: **end for**

3.1.2 Distinct Slice Deployment Algorithms

The main objective of slice orchestration is minimizing the deployment cost on the premise of meeting slice requirements. Considering the differentiated demands of three use case families defined in ITU, three corresponding types of slices have their peculiar objectives in addition to the main objective. No matter which kind of slices are required, the ultimate goal is to take advantage of infrastructure resources

efficiently. Hence, the main objective can be expressed by

$$\min \left[\sum_{n_R \in N_R} C_R\left(n_R\right) \cdot \mu_{n_R, n_I} + \sum_{e_R \in E_R} B_R\left(e_R\right) \cdot \nu_{e_R, e_I} \right]. \tag{3.8}$$

The eMBB usage scenario covers a range of cases, including wide-area coverage and hotspot. For the hotspot case, i.e., for an area with high user density, very high traffic capacity is needed, while the requirement for mobility is low and user data rate is higher. This kind of slice does not require strict delay and plentiful resources. Hence, the deployment objective of eMBB slices should be maximizing the remaining resources of physical nodes, which can be represented by

$$\max \left[\sum_{n_I \in N_I} C_I\left(n_I\right) - \sum_{n_R \in N_R} C_R\left(n_R\right) \cdot \mu_{n_R, n_I} \right]. \tag{3.9}$$

The mMTC usage scenario is characterized by a very large number of connected devices typically transmitting a relatively low volume of non-delay sensitive data. This use case has plenty of connections, which results in the requirement of high computing resources and low congestion rate. Therefore, the deployment objective should be minimizing the usage of bandwidth on physical links. In other words, the remaining bandwidth on physical links should be maximized. Thus, the deployment objective of mMTC slice can be denoted as

$$\max \left[\sum_{e_I \in E_I} B_I\left(e_I\right) - \sum_{e_R \in E_R} B_R\left(e_R\right) \cdot \nu_{e_R, e_I} \right]. \tag{3.10}$$

The uRLLC usage scenario has stringent requirements for capabilities such as throughput, latency, and availability. Some examples include wireless control of industrial manufacturing, remote medical surgery, transportation safety, etc. The QoS guarantee of this use case is low latency, which causes that the deployment objective should be minimizing the delay of slices. We transfer delay time into the number of hops, so minimizing the delay means minimizing each physical path length. Hence, deployment objective of uRLLC slices is

$$\min \sum_{e_R^{wl} \in E_R^{wl}} \nu_{e_R^{wl}, e_I^{wl}} + \sum_{e_R^{ol} \in E_R^{ol}} \nu_{e_R^{ol}, e_I^{ol}}. \tag{3.11}$$

These objectives are subject to

$$\sum_{n_R^{BS} \in N_R^{BS}} \mu_{n_R^{BS}, n_I^{BS}} = 1, \forall n_I^{BS} \in N_I^{BS}. \tag{3.12}$$

$$\sum_{n_I^{BS} \in N_I^{BS}} \mu_{n_R^{BS}, n_I^{BS}} \leq 1, \forall n_R^{BS} \in N_R^{BS} \tag{3.13}$$

Equation (3.12) ensures that each virtual BS only should be mapped to a physical BS. Equation (3.13) ensures that each physical BS only should undertake a virtual BS for each NSR.

$$\sum_{e_I^{wl} \in E_I^{wl}} B_I^{wl}\left(e_I^{wl}\right) \leq B_I^{wl} \tag{3.14}$$

$$\sum_{e_I^{wl} \in E_I^{wl}\left(n_I^{BS}\right)} C_I^{wl}\left(e_I^{wl}\right) \leq C_I^{BS}\left(n_I^{BS}\right), \forall n_R^{BS} \in N_R^{BS} \tag{3.15}$$

$$\sum_{n_R^{BS} \in N_R^{BS}} \mu_{n_R^{BS}, n_I^{BS}} \cdot C_R^{BS}\left(n_R^{BS}\right) \leq C_I^{BS}\left(n_I^{BS}\right), \forall n_I^{BS} \in N_I^{BS} \tag{3.16}$$

$$\sum_{n_I^{BS} \in N_I^{BS}} \sum_{e_I^{wl} \in E_I^{wl}\left(n_I^{BS}\right)} \mu_{n_R^{BS}, n_I^{BS}} \cdot \xi_{n_R^{BS}, e_I^{wl}} \cdot C_I^{wl}\left(e_I^{wl}\right) \geq C_R^{BS}\left(n_R^{BS}\right), \forall n_R^{BS} \in N_R^{BS}. \tag{3.17}$$

Equation (3.14) ensures that the bandwidth occupied by all the wireless channels should not exceed the total available bandwidth B_I^{wl} for each BS. Equation (3.15) ensures that the channel capacity sum of wireless links in each BS should not exceed capacity of this BS allocated by the control plane. Equation (3.16) ensures that the capacity sum of all virtual BS undertaken in this BS should not exceed its allocated capacity. Equation (3.17) ensures that the allocated capacity for each virtual BS should not be less than its capacity requirement.

$$\sum_{n_R^{OS} \in N_R^{OS}} \mu_{n_R^{OS}, n_I^{OS}} = 1, \forall n_I^{OS} \in N_I^{OS} \tag{3.18}$$

$$\sum_{n_I^{OS} \in N_I^{OS}} \mu_{n_R^{OS}, n_I^{OS}} \leq 1, \forall n_R^{OS} \in N_R^{OS} \tag{3.19}$$

Equation (3.18) ensures that each virtual OS only should be mapped to a physical OS. Equation (3.19) ensures that each physical OS only can undertake a virtual OS for each NSR.

$$\sum_{n_R^{CON} \in N_R^{CON}} \mu_{n_R^{CON}, n_I^{CON}} = 1, \forall n_I^{CON} \in N_I^{CON} \tag{3.20}$$

$$\sum_{n_I^{CON} \in N_I^{CON}} \mu_{n_R^{CON}, n_I^{CON}} \leq 1, \forall n_R^{CON} \in N_R^{CON} \tag{3.21}$$

$$\sum_{n_R^{CON} \in N_R^{CON}} \mu_{n_R^{CON}, n_I^{CON}} \cdot C_R^{CON}\left(n_R^{CON}\right) \leq C_I^{CON}\left(n_I^{CON}\right), \forall n_I^{CON} \in N_I^{CON} \tag{3.22}$$

$$\sum_{n_I^{CON} \in N_I^{CON}} \mu_{n_R^{CON}, n_I^{CON}} \cdot C_I^{CON}\left(n_I^{CON}\right) \geq C_R^{CON}\left(n_R^{CON}\right), \forall n_R^{CON} \in N_R^{CON} \tag{3.23}$$

Equation (3.20) ensures that each virtual core nodes only should be mapped to a physical servers. Equation (3.21) ensures that each physical server only can undertake a virtual core node for each NSR. Equation (3.22) ensures that the computing resource of each physical server can satisfy the total requirement of all virtual core nodes mapped in it. Equation (3.23) ensures that the computing resource of selected physical server should not be less than the computing resource requirement of virtual core nodes.

Three different deployment and orchestration strategies are provided to these three types of slices according to their objectives, respectively. After the arriving of NSRs, these requests are classified, then implemented by different mapping algorithms, respectively. Meanwhile, the resource efficiency (RE) and acceptance ratio (AR) of NSRs are calculated. Resource efficiency is defined as the revenues and cost ratio. The achieved revenues of accepting a NSR by the infrastructure network can be defined as the sum of nodes capacity and link bandwidth requirements of a NSR. And the cost can be defined as the sum of nodes capacity and link bandwidth resources of the infrastructure network. Hence, the resource efficiency can be formulated as follows:

$$RE = \frac{\sum_{n \in N_R} C_R(n) + \sum_{l \in E_R} B_R(l)}{\sum_{n \in N_R} C_R(n) + \sum_{l \in E_R} B_R(l) \times hop(l)}, \tag{3.24}$$

where $C_R(n)$ represents the capacity of node n and $B_R(l)$ represents the bandwidth of link l, $hop(l)$ represents the mapping path length of link l. Furthermore, acceptance ratio is the ratio of the number of NSRs which have been successfully mapped and the total number. Hence, it can be formulated as

$$AR = \frac{\sum_{t=0}^{T} NUM_{acc}}{\sum_{t=0}^{T} NUM_{arr}}. \tag{3.25}$$

In the above formula, NUM_{acc} represents the number of NSRs that have been accepted while NUM_{arr} denotes the number of NSRs that have been arrived. Details are presented in Algorithm 6.

Algorithm 6 The NSRs implementation algorithm

Require: $G_I = (N_I, E_I, C_I, B_I)$ and $R_{NS} = R_e \cup R_m \cup R_u$
Ensure: AR, RE, M_{node} and M_{link}
while $R_{NS} \neq \emptyset$ **do**
  Calculating the resources of infrastructure network G_I.
  if NSR is eMBB slice R_e **then**
    deploy it using Algorithm 7.
  else if NSR is mMTC slice R_m **then**
    deploy it using Algorithm 8.
  else
    deploy it using Algorithm 9.
  end if
  if M_{node} and M_{link} are not null **then**
    update the resources of infrastructure network G_I.
    calculating resource efficiency RE.
  else if M_{node} and M_{link} are null **then**
    calculating acceptance ratio AR.
  end if
end while

In the deployment algorithm for eMBB slice, the virtual BSs are first sorted and mapped according to the node mapping algorithm (Algorithm 4). After mapping the virtual BSs, the virtual CONs are mapped similarly considering the computing resource requirements. Then, the mapping of virtual OSs are finished when searching the shortest paths between each BS-CON pair. Finally, the virtual links are mapped with the link mapping algorithm (Algorithm 5). Details are shown in Algorithm 7.

Algorithm 7 Deployment algorithm A for eMBB slice

Require: G_I and R_e
Ensure: M_{node} and M_{link}
sort the virtual nodes by Algorithm 3.
do node mapping of BSs by Algorithm 4.
do node mapping of CONs by Algorithm 4.
do link mapping by Algorithm 5.
do node mapping of OSs according to the link mapping.
return the mapping results.

In the deployment algorithm for mMTC slice, the virtual CONs are mapped firstly. Next, take these CONs as the source endpoints and find out the candidate BSs as the target endpoints. Then, we select the shortest path from the set of candidate paths between CON and candidate BSs to map the virtual link. Details are shown in Algorithm 8.

In the deployment algorithm for uRLLC slice, we first find out the set of candidate BS-CON pairs and search all the possible routing paths between these candidate pairs as the candidate path set. According to the number of virtual links

Algorithm 8 Deployment algorithm B for mMTC slice

Require: G_I and R_m
Ensure: M_{node} and M_{link}
1: sort the virtual nodes by Algorithm 3.
2: do node mapping of CONs by Algorithm 4.
3: select available BSs as candidate BSs.
4: search the set of candidate paths between CON and candidate BSs.
5: do link mapping based on candidate paths by Algorithm 5.
6: do node mapping of OSs according to the link mapping.
7: do node mapping of BSs.
8: return the mapping results.

of NSR, we select the shortest paths from the set of candidate paths to map the virtual links. Then we map the virtual BSs and CONs into the physical endpoints of the selected paths. Details are shown in Algorithm 9.

Algorithm 9 Deployment algorithm C for uRLLC slice

Require: G_I and R_u
Ensure: M_{node} and M_{link}
1: sort the virtual nodes by Algorithm 3.
2: select available BSs as candidate BSs.
3: select available CONs as candidate CONs.
4: search the set of candidate paths between candidate CON and candidate BS.
5: do link mapping based on candidate paths by Algorithm 5.
6: do node mapping of OSs according to the link mapping.
7: do node mapping of BSs.
8: do node mapping of CONs.
9: return the mapping results.

3.2 Real-Time Slice Orchestration Framework

6G is expected to satisfy the dynamic and differentiated demands of users through real-time micro-management of multiple resources including communication, computing and storage resources. As the enablers of network slicing, NFV decouples software and hardware by virtualizing network functions and running them on the virtual machines (VMs) while SDN architecture provides centralized control plane for the configuration of network resources. These techniques prompt a service-based E2E wireless network architecture where VNFs of RANs and core network are placed as VMs deployed in data centers (DCs) of cloud InPs. The diverse demands of tenants can be satisfied through flexibly managing resources and efficiently orchestrating VNFs of slices. By involving tenants in VNE calculation, virtual networks could be provided in a tenant-driven manner with a trade-off

between cost-effectiveness and time-efficiency. Since E2E slices require multiple resources, multiple domains administrated by different cloud InPs form a federated environment to jointly provide tenants with resources.

E2E network slicing across multiple infrastructures has been discussed in the literature while MANO operations of slices in multiple administrative domains are also concern [15], as well as the life-cycle management operations. Through flexible slicing, heterogeneous resources of these cloud infrastructures can be utilized in a customized manner and the additional costs of the coalition can be reduced [17]. Besides, analyzing the profit of resources provisioning and monitoring the status of resource utilization are essential in dynamic real-time E2E slicing. Specifically, performing admission control of resource requests needs to consider the revenue of NSPs as well as the service requirements and reallocating resources across multiple domains requires a global view of slice deployment status. To handle massive requests of configuring and modifying E2E slices dynamically, RL methods are used in the real-time slice orchestration framework, improving the speed and accuracy of decision-making.

3.2.1 Hierarchical Slice Orchestration Architecture

Planning deployment location from a global view can effectively avoid resource competition caused by the increasing number of co-located VMs on the same server. Scheduling multi-dimensional resources in a comprehensive and balanced way can potentially increase resource efficiency and avoid resource waste and shortage. In order to improve the revenue of resource providers, managing multi-domain resources centrally and allocating optimal amount resources to E2E slice instances are required. Given that traffic load variation of the slice might degrade QoE, the centralized management approach faces performance issues and limits the autonomy of the tenants. Hence, based on the MANO architecture for multi-domain slices in 5G [15], an AI-based hierarchical resource management framework shown in Fig. 3.1 is proposed to integrate intelligence in customized slicing for 6G use cases in the scenario of multi-InPs and multi-tenant.

To meet dynamically evolving service quality requirements and support fine-grained network decision optimization, the proposed framework introduces a global resource manager (GRM) to handle incoming differentiated resource requests from tenants, and multiple local resource manager (LRM) to deal with the demand changes in resource requirements for individual tenant. The deployment of GRM and LRMs enables two-layer customization of slices, which means that the resources are firstly allocated to each tenant according to the heterogeneous slice performance requirements, and then resource allocation to each slice is optimized and adjusted according to the real-time observation of demand changes. It is worth noting that the AI-based algorithms used in global resource allocation and local slice adaption can be different.

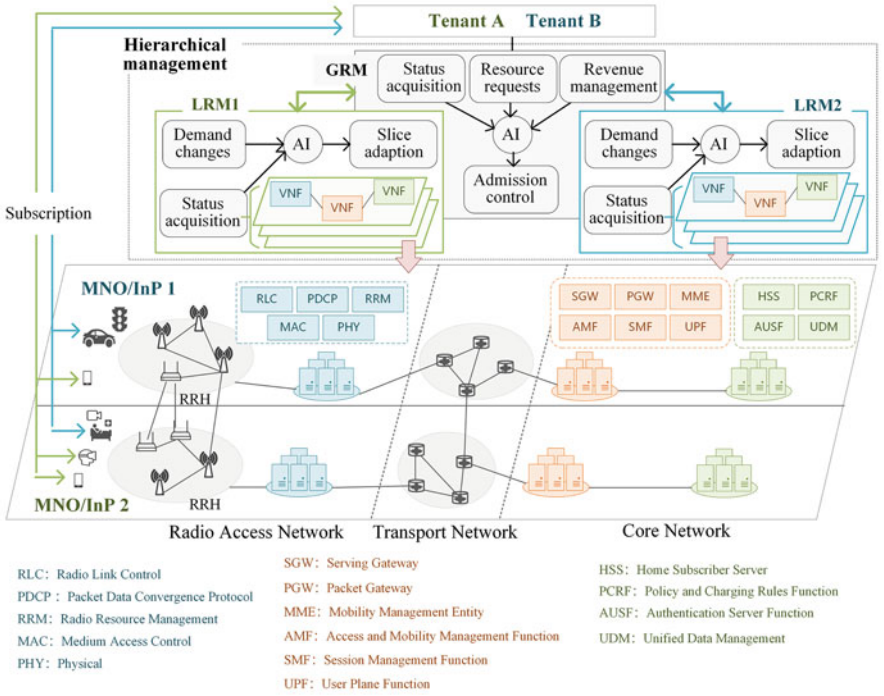


Fig. 3.1 The AI-based hierarchical slice orchestration architecture

The hierarchical approach can enable flexibility and scalability properties by distributing resource management to individual tenant. GRM maintains the overall control over the LRMs and delegates the concrete operations to each LRM. GRM is responsible for charging of slice owners and monitoring the LRMs while allocating federated resources across multiple domains. LRM performs slice adaption by adjusting the assigned resources to maintain service quality. Moreover, the LRMs not only provide each tenant the ability of resource customization but also have the distinguishing feature of transmitting the status of slices to the GRM.

To handle with the traffic dynamics quickly, the status of slices which include the deployment location of VNFs and the condition of traffic flows passing through these VNFs are observed periodically. Monitoring slice deployment and resource utilization facilitates to maintain service quality and enhance resource efficiency. Observing the real-time status of E2E slices provides a reference for determining whether or not to perform resource adaption. To realize real-time resource monitoring and slice topology information updating in the scenario of multi-InPs and multi-tenant, both of the differentiated slices provided by multiple tenants and the joint infrastructure network which consists of multiple infrastructures are depicted. Specifically, the cooperation between these infrastructure networks and the mapping relationships between the physical servers and the VNFs deployed in these servers are precisely delineated.

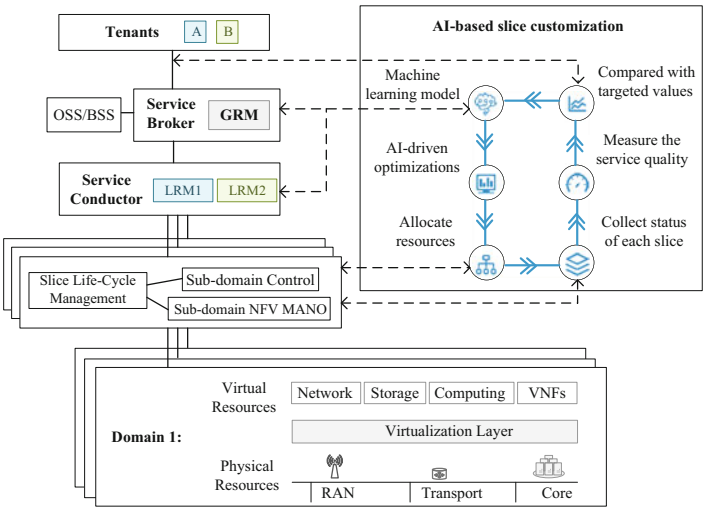


Fig. 3.2 The AI-enabled slice orchestration mechanism

3.2.2 AI-Enabled Slice Orchestration Mechanism

Figure 3.2 shows the procedure of customized slicing with the proposed AI-based management framework. After receiving the real-time slice requests from multiple tenants, the GRM deployed in the functional plane named the Service Broker performs admission control of these requests based on ML model. The NSPs make a trade-off between the resource requirements associated with these requests and the revenue achieved by providing required resources. Multi-dimensional resources are allocated to tenants with the objective of maximizing the long-term revenue of NSPs. For the DRL-based resource allocation performed in GRM, states are defined as the number of accepted requests belonging to different tenants, actions taken by each agent are accepting/rejecting the arrival slice requests and reward is related to slice utility. With the output of the DRL algorithm in GRM, slices are deployed and the status of slices are recorded periodically.

Depending on perceived status, the current service quality can be measured and compared with its target quality requirements. The current service quality satisfaction reflects the gap with the target value. The target values regarded as the desired service quality should be defined as the level of service that the available resources of InPs can and should provide, thus they are preset and fed as input to the optimization problem of slice adaption. As the slice-level feedback, current service quality satisfaction is used to improve the performance of ML model and update the model with demand changes that could occur over time. When there are changes in slice requests, such as a sudden increase in resource requirements, the ML model is utilized to maintain service quality for admitted slices by micro-managing resources.

The motivation of performing resource adaption generates from the mismatch between available resources and the varying traffic demand in the slice. This mismatch might cause two kinds of issues, one is that available resources are exhausted and the other is that partial resources are idle. The former means unfair resource allocation resulting in the low data rate of newly accepted user, and the latter means that the revenue of tenant is declining. To avoid unbalanced distribution of available resources, i.e., some resources are under-utilized, some are over-utilized, the allocated resource of each tenant should be adjusted to maximize the profits of available resources. After receiving the requests of adjusting resources for multiple slices, tenant makes decisions by weighing the cost and revenue of adjusting resources for each slice to ensure optimal resource efficiency.

The LRM deployed in Service Conductor performs the DRL-based slice adaption. States are tied to the current service quality satisfaction and actions denotes whether slice adaption is permitted. Reward is defined as the revenue obtained by adjusting resource minus the resource consumption cost and operational cost. The revenue is related to the amount of money paid by the service subscribers for guaranteeing service quality, which depends on the type of slice. The resource consumption cost represents the cost of providing more resources, such as the extra processing units required by the newly arrived traffic flows. The operational cost means the cost of performing reconfiguration, which includes the cost of service interruption caused by reallocating resources and migrating VNFs among physical servers. There is no doubt that DQL used in this chapter can be replaced by other advanced DQL-based algorithms to achieve better performance.

3.3 Fast Slice Reconfiguration Solution

As the demands of customized services vary dynamically over time, failing to satisfy changing resource requirements of slices might degrade the QoS and compromise the revenues of SPs. In order to maintain high revenue of SPs and high quality of services, it is necessary to meet the changing resource requirements of slices in a timely manner at a lower cost. In the dynamic environment, the parameters of resources slicing need to be updated and the slice reconfiguration must be necessary to avoid requests rejections [10]. There are two main challenges in achieving the goals of optimal slice reconfiguration and fast resources reallocation. First, from the perspective of maximizing the long-term revenue of SPs, real-time slice reconfiguration decisions should consider the cost and revenue of satisfying the additional resource requirements. Since the user demands are dynamic and uncertain, the ability to predict the increase in resource requirements for a particular type of slice is essential to the optimization of resource utilization [12]. Second, slice requests are diversified for different use cases and slices require multi-type resources including computing and storage resources. In addition, the infrastructure network consists of numerous DCs connected together, and the servers of DCs own computing and storage resources [12]. Thus, how to concurrently reallocate

computing and storage resources effectively and rapidly without modeling the complex dynamic environment is another challenge [10].

To deal with the aforementioned challenges, an optimal and fast slice reconfiguration (OFSR) framework is developed. A Markov renewal process (MRP) is introduced to predict the next change in resource requirements and the duration between changes based on the memory of past changes. With the prediction information, OFSR framework precalculates the revenue and cost of reconfiguring each slice. Instead of making decisions once a change occurs, the proposed prediction technique enables the SP to make decisions for different classes of slices periodically from the perspective of optimizing long-term revenue. Further, a Markov decision process (MDP) is used to model the jointly reallocation decision of computing and storage resources considering the uncertainty of resource requirements for diversified slices. To make the decisions of slice reconfiguration optimally and rapidly under the varying service demands, dueling deep Q-learning (DDQL) which has a single Q-network with two-stream Q-function, i.e., the state-value function and the advantage function, is adopted. Compared with the conventional Q-learning, the learning process of DDQL is speeded up by quickly identifying the best action without learning the effect of each action for each state [13].

Figure 3.3 shows the slice reconfiguration model, where there are three major players [3, 16]. *InP* is the owner of the network and provides physical resources. *Tenants* perceived as SPs request resources from InP to meet the service demands of *End Users (EUs)*. SPs issue slice requests to InP after receiving the service demands of EUs, and the change of service demands will affect the resource requirements of

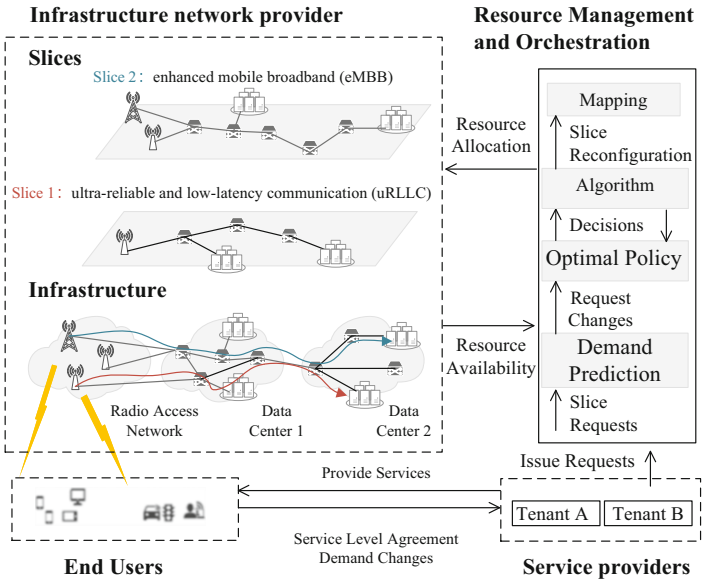


Fig. 3.3 The proposed framework of demand prediction based slice reconfiguration

slices. Guaranteeing the service quality of slices requires rapid response to changes in resource requirements, which means a timely decision on whether to permit slice reconfiguration. Hence, MRP-based demand prediction is considered in our system model of slice reconfiguration where the history of changes in resources requirements of slices is recorded and the next change is predicted. It means that the changes in resource requirements of each slice request and when the changes occur are recorded to predict the requirements of increasing resources. According to the prediction, SPs make decisions by weighing the cost and revenue of slice reconfiguration against the resource availability. Depending on the decisions of SPs, InP obtains revenue through providing extra resources for the demand changes.

In the slice reconfiguration model, there are two use cases provided by two different tenants separately, eMBB and uRLLC which own different characteristics in terms of SLA. The resource management and orchestration (RMO) block is responsible for managing virtual resources, orchestrating VNFs, and adjusting resource allocation to maintain the service quality of slices. After collecting slice requests, *Demand Prediction* component is in charge of analyzing the resource requirements and providing predicted information to *Optimal Policy* component. Then, *Optimal Policy* component makes decisions whether to do slice reconfiguration. *Algorithm* component is used to calculate the optimal policy and observe the results after the decision execution. The observation is applied for the training process as explained in Section IV. Hence, *Algorithm* component has the ability to efficiently deal with the uncertainty of changes by constantly learning from previous experiences. Once the reconfiguration request of a slice is permitted, RMO block will initiate the procedure of reallocating resources and rerouting the traffic flows. The goal of this procedure can be to minimize the routing paths between VNFs or maximize the remaining bandwidth resources of links, which depend on the characteristics of the service delivered by the slice. In our future work, we will provide detailed policies for VNFs migration and routing path re-planning.

3.3.1 A MRP Based Demand Prediction Model

To provide E2E services, InP possess RAN and multiple DCs which provide diverse resources [6], e.g., computing and storage resources. Each slice consists of VNFs distributed geographically in standard universal servers of DCs, and physical network functions in RAN. When a slice request arrives, InP will provide RMO all information including the topology and resource availability. With these information, the orchestrator will deploy VNFs in the optimal servers and the manager will allocate resources to each VNF. As for the resource reallocation of heterogeneous slice requests in the dynamic and uncertain environment, the allocated resources can be changed with the scaling-in and scaling-out mechanisms, and even the placement of VNFs can be migrated among different servers.

The infrastructure network is modeled as an undirected weighted graph $G(\mathcal{N}, \mathcal{L}, C, \mathcal{B})$, where $\mathcal{N}$ is the physical node set, $\mathcal{L}$ is the physical link set, C

is the capacity of nodes, and $\mathcal{B}$ is the bandwidth of links. Specifically, the servers are regarded as physical nodes and the physical links are the connections between servers. Different VNFs of the same slice can be deployed on multiple physical nodes, and a physical node can host multiple VNFs of different slices. C means a collection of computing and storage resources, and the maximum computing and storage resources of the infrastructure are represented by Θ and Ω units, respectively. Assuming that there are no restrictions in the bandwidth of links, which means that the bandwidth is always enough for demand changes.

Assuming that there are K classes of slices, denoted by $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$. Each slice of class k can be represented by $S\left(\mathcal{N}^k, \mathcal{L}^k, C^k, \mathcal{B}^k\right)$, where $\mathcal{N}^k$ and $\mathcal{L}^k$ denote the sets of virtual nodes and links, respectively. VNFs of slices are regarded as the virtual nodes, thus the virtual nodes set is a subset of physical nodes. C^k and $\mathcal{B}^k$, respectively, denote the capacity of virtual nodes and the bandwidth of virtual links. Assuming that the capacity of node n in a slice from class k contains δ_n^k units of computing resources and ω_n^k units of storage resources. Hence, the total node capacity of slice can be denoted as

$$C^k = \left\{ \sum_{n=1}^{\left|\mathcal{N}^k\right|} \delta_n^k, \sum_{n=1}^{\left|\mathcal{N}^k\right|} \omega_n^k \right\}, n \in \mathcal{N}^k, k \in \mathcal{K}, \tag{3.26}$$

where $\left|\mathcal{N}^k\right|$ is the number of nodes in a slice from class k. And the bandwidth of link l is denoted as β_l^k, thus

$$\mathcal{B}^k = \left\{ \sum_{l=1}^{\left|\mathcal{L}^k\right|} \beta_l^k \right\}, l \in \mathcal{L}^k, k \in \mathcal{K}, \tag{3.27}$$

where $\left|\mathcal{L}^k\right|$ is the number of links in a slice from class k. Let q^k denotes the number of slices from class k being served simultaneously. The following resource constraints guarantee that the allocated resources do not exceed the available resources of the infrastructure network:

$$\sum_{k=1}^{K} \sum_{n=1}^{\left|\mathcal{N}^k\right|} q^k \cdot \delta_n^k \leq \Theta, \sum_{k=1}^{K} \sum_{n=1}^{\left|\mathcal{N}^k\right|} q^k \cdot \omega_n^k \leq \Omega. \tag{3.28}$$

MRP is applied to achieve an efficiently accurate prediction of slices' resources evolution in the near future based on past resource consumption. An MRP is a two-dimensional stochastic process $(X_m, T_m)_{m \geq 0}$, and $(X_m)_{m \geq 0}$ is a Markov chain which represents the states successively visited [4]. If a stochastic process is an MRP, the semi-Markov kernel can be represented by

$$Q_{i,j}(t) = P\{X_{m+1} = j, T_{m+1} - T_m \leq t \,|X_m = i\}, \tag{3.29}$$

where X_m represent the state after the mth transition, T_m denotes the times at which the mth transitions occur. The sojourn time $T_{m+1} - T_m$ in any state depends on both the current state and the next transition.

$Q_{i,j}(t)$ denotes the probability that the process makes a transition from current state i into next state j within t units of time [1]. Further, the kernel can be rewritten as $Q_{i,j}(t) = P_{i,j}G_{i,j}(t)$, when $t \to \infty$, $P_{i,j} = \lim_{t\to\infty} Q_{i,j}(t)$. $P_{i,j}$ denotes the transition probability of states. $G_{i,j}(t)$ denotes the probability that the transition occurring in an amount of time t given that the process entered state i newly and will transfer to state j,

$$G_{i,j}(t) = P\{T_{m+1} - T_m \leq t \,|X_{m+1} = j, X_m = i\}. \tag{3.30}$$

As suggested in [1], estimating the likelihood of future transitions within a certain time window is more effective than predicting the time at which a transition could occur. Hence, given $Q_{i,j}(t)$, we calculate

$$Q_{i,j}(t - \Delta t, t + \Delta t) = Q_{i,j}(t + \Delta t) - Q_{i,j}(t - \Delta t), \tag{3.31}$$

where $Q_{i,j}(t \pm \Delta t)$ is the likelihood of making a transition from i to j within a time period t. The size of time window is $2\Delta t$, which is directly related to prediction accuracy.

Commonly, given the transition probability $P_{i,j}$, if $P_{i,o} = \max_{j\in\Phi_i}\{P_{i,j}\}$, where Φ_i represents the set of possible states which can be transitioned from state i, state o is most likely to be the next state of state i. Hence, the outcome of $P_{i,j}$ predictor is state o. Similarly, with the probability $Q_{i,j}(t)$ defined in Eq. (3.29), the results of MRP predictor will be such that $Q_{i,o}(t) = \max_{j\in\Phi_i}\{Q_{i,j}(t)\}$. Besides, the prediction accuracy can be improved by additionally considering the previous state, i.e., extending $Q_{i,j}(t)$ to $Q_{h,i,j}(t)$, which is represented by Eq. (3.32).

$$Q_{h,i,j}(t) = P\{X_{m+1} = j, T_{m+1} - T_m \leq t \,|X_m = i, X_{m-1} = h\}. \tag{3.32}$$

The diagram of the MRP-based prediction procedure is shown in Fig. 3.4. The database holds the record of demand changes, which provides information to compute $P_{i,j}$ and $G_{i,j}(t)$. With the arrival of new changes, $P_{i,j}$ and $G_{i,j}(t)$ can be periodically updated. Then $Q_{i,j}(t)$ is calculated and queried by the predictor to evaluate the probabilities. The inputs of the predictor include the current state i, the current sojurn time t_c of state i, and the time length t_l during which a transition is expected to occur. The output o is the next most likely state which occur at $t_c + t_l$. It is worth noting that there are both increment and decrement in resource requirement of slices. The increment means that SPs need to decide whether to provide more resources and reallocate them in order to guarantee the service quality

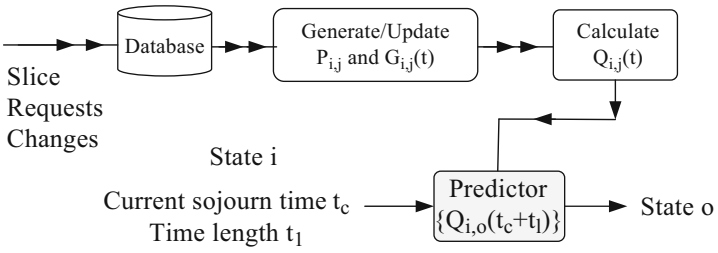


Fig. 3.4 MRP-based prediction procedure

of the slices. The decrement means the departure of data flows, which requires no resource consumption and does not affect service quality of slices. We assume that no resources are released during the life cycle of the slice, and all resources of this slice are released simultaneously when the service terminates. Hence, only the increment can trigger the proposed reconfiguration framework to make decision of slice reconfiguration.

Given the total remaining resources of the infrastructure network including Λ units of computing resources and Γ units of storage resources, $\Lambda \leq \Theta, \Gamma \leq \Omega$, the predicted increment of resource requirements can be used to determine if there are sufficient resources for slice reconfiguration. At a particular time, node n in a slice from class k requires extra computing resources $\Delta\delta_n^k$, storage resources $\Delta\omega_n^k$, and link l requires extra bandwidth $\Delta\beta_l^k$. Let p^k denotes the number of slices from class k being allowed to reconfigure simultaneously, $p^k \leq q^k$. The following resource constraints guarantee that the reallocated resources do not exceed the remaining resources,

$$\sum_{k=1}^{K} \sum_{n=1}^{\left|\mathcal{N}^k\right|} p^k \cdot \Delta\delta_n^k \leq \Lambda, \sum_{k=1}^{K} \sum_{n=1}^{\left|\mathcal{N}^k\right|} p^k \cdot \Delta\omega_n^k \leq \Gamma. \tag{3.33}$$

3.3.2 A DRL Based Slice Reconfiguration Policy

In order to handle the upcoming extra resource requirements of different classes of slices, the proposed OFSR framework provides slices reconfiguration decisions according to the predicted results of the MRP-based prediction procedure. With the aim of maximizing the revenue of SPs with minimal reconfiguration cost, the revenue minus cost of slice reconfiguration is calculated as the basis for decision-making. For SPs, the revenue is the increased user utility while the cost is additional resource consumption and service interruption caused by migrating VNFs and updating resources.

Table 3.3 Notation of description for slice reconfiguration

Symbol	Description
φ_n^c	Cost function of computing resource for node n
φ_n^s	Cost function of storage resource for node n
φ_l	Cost function of bandwidth resource for link l
σ_n	User utility weight of node capacity
σ_l	User utility weight of link bandwidth
$\boldsymbol{\theta}_{\mathbf{x}}$	Cost coefficient for updating resources of links
$\boldsymbol{\theta}_{\mathbf{y}}$	Cost coefficient for updating resources of VNFs
$\boldsymbol{\theta}_{\mathbf{z}}$	Cost coefficient for migrating VNFs
ϕ_u	Resource cost of slice from class k
ϕ_v	Service interruption cost of slice from class k
U_k	User utility in slice from class k

For each request of satisfying additional resource requirements, SPs estimate whether the locations of VNFs need to be migrated. Migrating VNFs tends to introduce higher reconfiguration overhead than only updating resources because of VNFs instantiation. Assuming that both of the signaling cost and the communication overhead are relatively low and negligible. Hence, the reconfiguration cost includes two parts, i.e., the service interruption caused by migrating VNFs and updating resources, the cost of offering more computing and storage resource to VNFs, and bandwidth resource to links. The reconfiguration description notations are summarized in Table 3.3.

With the pricing scheme of resources in [19], the resource cost ϕ_u is a function of the extra required resources $\left(\Delta\delta_n^k, \Delta\omega_n^k, \Delta\beta_l^k\right)$. The resource cost function can be presented as φ_n^c, φ_n^s, φ_l, and thus we have Eq. (3.34).

$$\begin{aligned}\phi_u\left(\Delta\delta_n^k, \Delta\omega_n^k, \Delta\beta_l^k\right) = &\sum_{n\in N^k}\varphi_n^c\left(\Delta\delta_n^k\right)+\\ &\sum_{n\in N^k}\varphi_n^s\left(\Delta\omega_n^k\right)+\sum_{l\in L^k}\varphi_l\left(\Delta\beta_l^k\right).\end{aligned} \tag{3.34}$$

The cost of service interruption is related to the time required for state transition, which is a function of the state difference of the links and VNFs before and after reconfiguring the slice. The state difference of links are reflected by variables $\mathbf{x}$ and the state difference of VNFs include two cases, updating the resources of VNFs only, and migrating VNFs while updating the VNFs capacity. Updating the resources of VNFs is reflected by variables $\mathbf{y}$ while migrating VNFs is reflected by variables $\mathbf{z}$. Hence, for the first case,

$$\phi_v = \boldsymbol{\theta}_{\mathbf{x}}^T \cdot I\left(\mathbf{x}-\mathbf{x}_0\right) + \boldsymbol{\theta}_{\mathbf{y}}^T \cdot I\left(\mathbf{y}-\mathbf{y}_0\right). \tag{3.35}$$

For the second case,

$$\phi_v = \boldsymbol{\theta}_{\mathbf{x}}^T \cdot I\left(\mathbf{x} - \mathbf{x}_0\right) + \boldsymbol{\theta}_{\mathbf{y}}^T \cdot I\left(\mathbf{y} - \mathbf{y}_0\right) + \boldsymbol{\theta}_{\mathbf{z}}^T \cdot I\left(\mathbf{z} - \mathbf{z}_0\right), \tag{3.36}$$

where $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0)$ are the states before reconfiguration, $\boldsymbol{\theta}_{\mathbf{x}}, \boldsymbol{\theta}_{\mathbf{y}}, \boldsymbol{\theta}_{\mathbf{z}}$ are the cost coefficient and $I(\cdot)$ is an indicator function, i.e., if $x \neq 0$, $I(x) = 1$; otherwise, $I(x) = 0$. The change in bandwidth allocation of links $\mathbf{x} - \mathbf{x}_0$ accounts for the cost of control signaling used to change the routing path of data flows. The change in updating the resources of VNFs $\mathbf{y} - \mathbf{y}_0$ accounts for the cost of adjusting the capacity of VNFs while the change in migrating VNFs $\mathbf{z} - \mathbf{z}_0$ accounts for the cost of changing the schedule of VNFs. The reconfiguration cost coefficient depends on how many units of resources will be consumed to perform a single reconfiguration, which might be different on different platforms and vary with slice classes. Note that migrating VNFs requires the reconfiguration of the physical servers while adjusting the capacity of VNFs mainly involves the signaling overhead, thus migrating VNFs requires higher cost.

The revenue of reconfiguring slice is defined as the user utility because SPs make profit by serving EUs. Higher utility means that a slice could make more profit, and more reconfigurations could be tolerated by the slice [18]. In this paper, the user utility U_k in slice from class k is defined as the weighted sum of node capacity and link bandwidth. Hence, the increased user utility can be represented as

$$U_k = \sum_{n \in \mathcal{N}^k} \sigma_n \left(\Delta\delta_n^k + \Delta\omega_n^k\right) + \sum_{l \in \mathcal{L}^k} \sigma_l \left(\Delta\beta_l^k\right), \tag{3.37}$$

where σ_n and σ_l are the user utility weight of node capacity and link bandwidth, respectively. It is worth noting that both the cost and revenue of reconfiguring slices take into account the reallocation of bandwidth resources, which means that the proposed approach is also suitable for the scenario of constrained bandwidth. When bandwidth resources are limited, not only the VNFs placement and bandwidth resource occupancy should be perceived in real time but also the rerouting algorithm of traffic flows should be determined.

For slice o from class k which has additional resource requirements, the net revenue from reconfiguration is defined as r_o^k, $r_o^k = U_k - \phi$. With the resource constraints defined in Eqs. (3.28) and (3.33), the objective of slice reconfiguration is to find the optimal decision which maximizes the increased user utility U_k with minimal reconfiguration cost ϕ, $\phi = \phi_u + \phi_v$. SPs tend to reconfiguring slices that have a positive net revenue in decision-making, and the optimization objective is denoted as $\max \sum_k \sum_o r_o^k$. To make the decisions optimally for reconfiguration of diversified slices, the MDP is recruited [9]. At any time, each decision relies on the current state and the decision turns into a completely new stochastic state at next time. Consequently, the reconfiguration issue is formulated as a MDP defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ captures the state transition probabilities, and the state sojourn time and $\mathcal{R}$ is the reward function.

3.3.2.1 State Space

Recall that the proposed OFSR framework works in a proactive fashion: it makes decisions mainly according to the current resource usage of diversified slices and the information about the upcoming extra resource requirements. The state space $\mathcal{S}$ can be defined as Eq. (3.38).

$$\mathcal{S} \triangleq \{\mathbf{S}_t\} = \left(\chi_{oc}, \chi_{re}, \chi_{pr}, \chi_{ex}\right). \tag{3.38}$$

- χ_{oc} indicates the resources occupy status of q^k slices from class k ($\forall k \in \mathcal{K}$) in time step t. For slice o from class k, its resources occupy status can be defined as Eq. (3.39).
- χ_{re} indicates the remaining resources of all DCs in the infrastructure network, which can be denoted as Eq. (3.40).
- χ_{pr} indicates the predicted information of demand changes, which can be denoted as Eq. (3.41). If a slice o from class k will require more resources according to the output of the prediction procedure, $c_o^k = 1$, and if not, $c_o^k = 0$. Thus, let $O^k = \sum_{o=1}^{q^k} c_o^k$ denotes how many changes there are and then $p^k \le O^k \le q^k$.
- χ_{ex} indicates the extra resource requirements of q^k slices from class k ($\forall k \in \mathcal{K}$) in time step t. For a slice o from class k, its extra resources requirements can be defined as Eq. (3.42). If no demand changes occur in the slice o from class k, the extra required resources $\Delta\delta_{o,n}^k$, $\Delta\omega_{o,n}^k$, and $\Delta\beta_{o,l}^k$ are equal to 0.

$$\chi_{oc} = \left\{\left[\delta_{o,n}^k, \omega_{o,n}^k, \beta_{o,l}^k\right], n \in N^k, l \in L^k, 1 \le o \le q^k\right\} \tag{3.39}$$

$$\chi_{re} = \{[\delta_n, \omega_n, \beta_l], n \in \mathcal{N}, l \in \mathcal{L}\} \tag{3.40}$$

$$\chi_{pr} = \left\{\left[c_1^k, \ldots, c_o^k, \ldots, c_{q^k}^k\right], 1 \le o \le q^k\right\} \tag{3.41}$$

$$\chi_{ex} = \left\{\left[\Delta\delta_{o,n}^k, \Delta\omega_{o,n}^k, \Delta\beta_{o,l}^k\right], n \in N^k, l \in L^k, 1 \le o \le q^k\right\}. \tag{3.42}$$

3.3.2.2 Action Space

At state $\mathbf{S}_t$, for each slice $c_o^k = 1$, the SP determines whether to reconfigure this slice to maximize the long-term revenue. Let a_o^k denote the action to be taken at state $\mathbf{S}_t$, $a_o^k = 1$ if a slice o from class k with demand changes is reconfigured and $a_o^k = 0$ if this reconfiguration is not permitted. The state-dependent action space $\mathcal{A}$ can be

defined as Eq. (3.43). Hence, we also can obtain $p^k = \sum_{o=1}^{O^k} a_o^k$ from the action.

$$\mathcal{A}_s \triangleq \{a_s\} = \left\{ \left[a_1^k, \ldots, a_o^k, \ldots, a_{O^k}^k \right], 1 \leq o \leq O^k, k \in K \right\} \tag{3.43}$$

3.3.2.3 State Transition Probability

The transition to next state $\mathbf{S}_{t+1}$ is only determined by current state $\mathbf{S}_t$ and action performed. And the information about the transition probability is difficult to get in the dynamic environment. The transition from $\mathbf{S}_t$ to $\mathbf{S}_{t+1}$ with reward r_t when action a_s is taken can be characterized by the conditional transition probability, $P(\mathbf{S}_{t+1}, r_t | \mathbf{S}_t, a_s)$. Thus, the transition probability function can be denoted as Eq. (3.44).

$$\mathcal{P}_{\mathbf{SS}'}^a = P\left(\mathbf{S}_{t+1} = \mathbf{S}', r_t | \mathbf{S}_t = \mathbf{S}, a_s\right). \tag{3.44}$$

In this work, the transitions on χ_{pr} and χ_{ex} can be derived from the database created in the MRP-based prediction method while the transitions on χ_{oc} and χ_{re} depend on the previous reconfiguration decisions. The optimal policy that maximizes the revenue of SPs is learned through the RL algorithm by interacting with the environment in a try-and-error fashion without exact state transition probability.

3.3.2.4 Reward Function

$$r(\mathbf{S}_t, a_s) = \sum_{k=1}^{K} \sum_{o=1}^{O^k} r_o^k a_o^k \tag{3.45}$$

The reward in taking action a_s at state $\mathbf{S}_t$ is defined as Eq. (3.45). If slice o from class k is permitted to reconfigure, i.e., $a_o^k = 1$, the SP receives a reward r_o^k. Otherwise, if it is not permitted or there are no changes, the reward is equal to 0. After making the reconfiguration decisions, the current state will transit to next state $\mathbf{S}_{t+1}$. Based on the MDP model, the RL agent learns the best decision policy through maximizing the rewards in the interaction with its environment over time [14]. A decision policy π, i.e., reconfigure a slice or not, is made up of a series of consequent actions. The policy $\pi(\mathbf{S}_t, a_s)$ is a mapping from state space to action space $\mathcal{S} \rightarrow \mathcal{A}$, which is equal to the probability of taking action a_s conditioned on the current state $\mathbf{S}_t$.

According to Eq. (3.44), the policy function must satisfy $\sum_{a_s \in \mathcal{A}} \pi(\mathbf{S}_t, a_s) = 1$. The goal of RL is to learn an optimal policy to maximize the cumulative expected

rewards. Thus, the long-term cumulative discounted reward starting from state $\mathbf{S}_t$ at time t can be formulated as Eq. (3.46), where $\gamma \in (0, 1]$ is the discount factor to balance the importance of current rewards and the future rewards. Our objective is to find the optimal policy π^* that maximizes the reward $\mathcal{R}_\pi$, i.e., $\pi^* = \arg\max_\pi \mathcal{R}_\pi$. The objects of reconfiguration obviously are slices that have already been deployed, thus dealing with new slices is not taken into consideration in this work. However, in the case of considering the arrival and departure of new slices or the case of considering more resource types and slice types, the above formulations can be straightforwardly extended by accommodating additional states to the state space of the current model. The action space will be kept the same and the rewards of the actions need to be reset in the problem formulations.

$$\mathcal{R}_\pi\left(\mathbf{S}_t\right) = \sum_{\tau=0}^{\infty} \gamma^\tau r\left(\mathbf{S}_t, \pi\left(\mathbf{S}_t\right)\right), \forall \mathbf{S}_t \in \mathcal{S} \tag{3.46}$$

Because of the high complexity of the optimization problem and the uncertainty of changes in resource requirements, the DDQL algorithm is used in our proposed slice reconfiguration framework to deal with the large state space and multi-dimension reconfiguration decisions. The classical Q-learning algorithm is one of the widely used RL technique, learns from its decisions, and adjusts its policy to converge to the optimal policy after a finite number of iterations [21]. It constructs a lookup table storing all state-action values, and the entry of this lookup table is initialized arbitrarily. To avoid getting stuck at non-optimal policies, ϵ-greedy algorithm is often used to select action. This algorithm introduces a parameter ϵ which suggests for the agent in taking a random action with probability ϵ or taking the action a^* that maximizes lookup table value with probability $1 - \epsilon$ for each time step. After acquiring a new experience as a result of the chosen action, the Q-learning algorithm updates the lookup table entry based on the updating rule denoted as Eq. (3.47). $\mathbf{S}'$ is the next state, a' is the action at $\mathbf{S}'$, and $\alpha_r \in (0, 1]$ is the learning rate.

$$Q\left(\mathbf{S}, a\right) := Q\left(\mathbf{S}, a\right) + \alpha_r \left[r\left(\mathbf{S}, a\right) + \gamma \max_{a'} Q\left(\mathbf{S}', a'\right) - Q\left(\mathbf{S}, a\right) \right] \tag{3.47}$$

Since the Q-learning algorithm fails in the convergence rate when the state space and action space are large, a deep Q-network $Q\left(\mathbf{S}, a; \theta_i\right)$ with parameters (weights) θ_i at iteration i is usually used for the high-dimensional environment. Deep Q-learning (DQL) algorithm which is developed by Google DeepMind [8] integrates DNN with RL. When a neural network is used to represent Q-function, the performance might be unstable because that a small update of Q-values may significantly affect the policy and therefore the data distribution, and the correlations between the Q-values and the target values $r\left(\mathbf{S}, a\right) + \gamma \max_{a'} Q\left(\mathbf{S}', a'\right)$. To address these instabilities, the experience replay mechanism is used in DQL to remove correlations in the observation sequence and smooth over changes in the data

distribution. Additionally, the target Q-network is used to periodically update Q-network so that the correlations between target values and estimated Q-values will be reduced.

To perform the experience replay, the experiences $e_t = (\mathbf{S}_t, a_s, r_t, \mathbf{S}_{t+1})$ are stored at each time step t in a dataset $\mathcal{D}_t = \{e_1, e_2, \ldots, e_t\}$. During the learning process, the mini-batches of experience $\left(\mathbf{S}, a, r, \mathbf{S}'\right) \sim U(\mathcal{D})$ will be sampled from the dataset to feed into the neural network. The DQL algorithm updates the neural network by optimizing the following loss function at iteration i:

$$L_i(\theta_i) = \mathbb{E}_{(\mathbf{S},a,r,\mathbf{S}')\sim U(\mathcal{D})}\left[\left(y_i^{DQN} - Q(\mathbf{S}, a; \theta_i)\right)^2\right] \tag{3.48}$$

with

$$y_i^{DQN} = r + \gamma \max_{a'} Q\left(\mathbf{S}', a'; \theta_i^-\right), \tag{3.49}$$

where θ_i^- is the network parameter used to compute the target at iteration i. The parameters of the target network $Q\left(\mathbf{S}', a'; \theta_i^-\right)$ are freezed for a fixed number of interactions while the Q-network $Q(\mathbf{S}, a; \theta_i)$ are updated by gradient descent. The gradient is defined as Eq. (3.50).

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{(\mathbf{S},a,r,\mathbf{S}')\sim U(\mathcal{D})}\left[\left(y_i^{DQN} - Q(\mathbf{S}, a; \theta_i)\right)\nabla_{\theta_i} Q(\mathbf{S}, a; \theta_i)\right] \tag{3.50}$$

Double Deep Q-learning (Double DQL) algorithm introduced to further improve the performance of DQL is the same, but with the target y_i^{DQN} replaced by y_i^{2DQN}.

$$y_i^{2DQN} = r + \gamma Q\left(\mathbf{S}', \arg\max_{a'} Q\left(\mathbf{S}', a'; \theta_i\right); \theta_i^-\right) \tag{3.51}$$

The key innovation is the usage of experience replay, which increases the efficiency of learning and enhances the stability of DNN. The agent of DQL with experience replay stores transitions that has been experienced for a period of time and reuses the minibatch data for many times to update the network parameters. In addition, experience replay alleviates the violation of independent and identically distributed assumption of training data, and uniform sampling from the dataset reduces the correlation among the samples used in the update.

Although the DQL algorithm performs greater than the traditional RL algorithms, there are still many literatures which focus on further improving the convergence speed and achieving higher stability. In this paper, the deep dueling network [20] is introduced in the proposed framework. Comparing to the standard neural networks, such as convolutional networks, the *dueling architecture* has two streams to estimate state values and action advantages separately instead of estimating the action-value function.

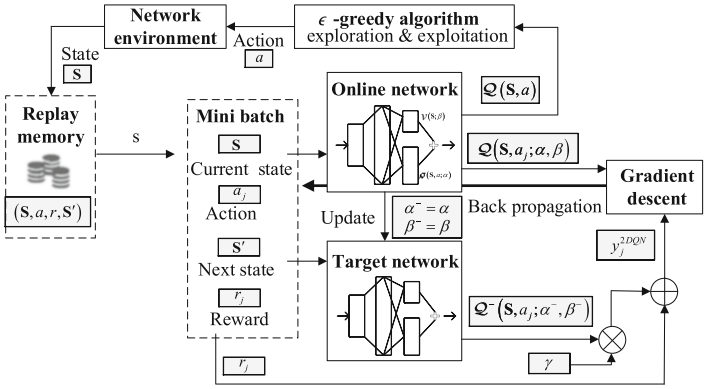


Fig. 3.5 The architecture of the DDQL algorithm

The two streams of fully connected layers are conducted to separately compute the value function and advantages function, one stream of fully connected layers in the dueling network outputs a scalar $\mathcal{V}(\mathbf{S};\beta)$ and the other stream outputs an $|\mathcal{A}|$-dimensional vector $\mathcal{G}(\mathbf{S},a;\alpha)$, which is shown in the Fig. 3.5. α and β are the parameters of the two streams of fully connected layers. With the definition of advantage function, the Q-function can be obtained by combining the two streams as follows:

$$\mathcal{Q}(\mathbf{S},a;\alpha,\beta) = \mathcal{V}(\mathbf{S};\beta) + \mathcal{G}(\mathbf{S},a;\alpha)\,. \tag{3.52}$$

Note that $\mathcal{Q}(\mathbf{S},a;\alpha,\beta)$ is only a parameterized estimate of the true Q-function. Given $\mathcal{Q}$, $\mathcal{V}$, and $\mathcal{G}$ cannot be obtained uniquely. To address this issue, the advantage function estimator is forced to have zero advantage when choosing action, which means that the last module of the network implements the forward mapping:

$$\mathcal{Q}(\mathbf{S},a;\alpha,\beta) = \mathcal{V}(\mathbf{S};\beta) + \left(\mathcal{G}(\mathbf{S},a;\alpha) - \max_{a'\in\mathcal{A}} \mathcal{G}(\mathbf{S},a';\alpha)\right). \tag{3.53}$$

The max operator in Eq. (3.53) can be replaced with an average, thus an alternative module can be denoted as Eq. (3.55). Although Eq. (3.55) loses the original semantics of $\mathcal{V}$ and $\mathcal{A}$, it increases the stability of the optimization [20]. Hence, the module of equation (3.55) is used in the deep dueling algorithm of the proposed OFSR framework. The details of the DDQL algorithm are shown in Algorithm 10. Compared with the static model-based algorithms which suffers from higher computational complexity, the computational complexity of Algorithm 10 is $O\left(\mathbb{H}^{\vartheta}\mathbb{N}_{\rho}\right)$. $\mathbb{H}^{\vartheta}$ denotes the number of hidden layers while $\mathbb{N}_{\rho}$ denotes the number of neurons. The computational complexity of Algorithm 10 depends on the number of state and action sets involved in the learning process [13].

Algorithm 10 Dueling deep Q-learning algorithm

1: Initialize the replay memory $\mathcal{D}$, and the target network replacement frequency F^-.
2: Initialize the *online network* $Q(\mathbf{S}, a; \alpha, \beta)$ with weight parameters α, β.
3: Initialize the *target network* $Q^-\left(\mathbf{S}, a; \alpha^-, \beta^-\right)$ with weight parameters α^-, β^-.
4: **for** episode $e \in \{1, 2, \ldots, T\}$ **do**
5: Choose an action a according to the ϵ-greedy algorithm.
6: Take action a, and observe reward r and next state $\mathbf{S}'$.
7: Add $\left(\mathbf{S}, a, r, \mathbf{S}'\right)$ into the replay memory $\mathcal{D}$.
8: Select random mini-batches $\left(\mathbf{S}, a_j, r_j, \mathbf{S}'\right)$ from the replay memory $\mathcal{D}$.
9: Calculating Q-function by combining the value function and advantage function as Eq. (3.55).
10: Set y_j^{2DQN} based on Eq. (3.51).

$$y_j^{2DQN} = r_j + \gamma Q^-\left(\mathbf{S}', \arg\max_{a_j'} Q^-\left(\mathbf{S}', a_j'\right); \alpha^-, \beta^-\right) \tag{3.54}$$

11: Perform the gradient descent step on $\left(y_j^{2DQN} - Q\left(\mathbf{S}, a_j; \alpha, \beta\right)\right)^2$.
12: In every F^- steps, reset Q^-=Q.
13: **end for**

$$Q(\mathbf{S}, a; \alpha, \beta) = \mathcal{V}(\mathbf{S}; \beta) + \left(\mathcal{G}(\mathbf{S}, a; \alpha) - \frac{1}{|\mathcal{A}|}\sum_{a'}\mathcal{G}\left(\mathbf{S}, a'; \alpha\right)\right). \tag{3.55}$$

In the proposed OFSR framework, we use the RMO block as the learner. Based on the state of the multi-type resources usage and the state of predicted resources increment for differentiated slices, optimal reconfiguration decisions for different classes of slices are obtained to achieve high long-term revenue of SPs. It should be pointed out that the initial deployment location of these running slice instances determine the state space, thus a learning process for a new policy will be initiated when the state space changes.

Acknowledgments If you want to include acknowledgments of assistance and the like at the end of an individual chapter please use the `acknowledgement` environment—it will automatically render Springer's preferred layout.

References

1. Abu-Ghazaleh, H., Alfa, A.S.: Application of mobility prediction in wireless networks using markov renewal theory. IEEE Trans. Veh. Technol. **59**(2), 788–802 (2009)
2. Bari, F., Chowdhury, S.R., Ahmed, R., Boutaba, R., Duarte, O.C.M.B.: Orchestrating virtualized network functions. IEEE Trans. Netw. Serv. Manag. **13**(4), 725–739 (2016)
3. Bega, D., Gramaglia, M., Banchs, A., Sciancalepore, V., Samdanis, K., Costa-Perez, X.: Optimising 5G infrastructure markets: The business of network slicing. In: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, pp. 1–9. IEEE (2017)

4. Cinlar, E.: Introduction to Stochastic Processes. Courier Corporation, North Chelmsford, MA (2013)
5. Guan, W., Wen, X., Wang, L., Lu, Z., Shen, Y.: A service-oriented deployment policy of end-to-end network slicing based on complex network theory. IEEE Access **6**, 19691–19701 (2018)
6. Halabian, H.: Distributed resource allocation optimization in 5G virtualized networks. IEEE J. Sel. Areas Commun. **37**(3), 627–642 (2019)
7. Herker, S., Khan, A., An, X.: Survey on survivable virtual network embedding problem and solutions. In: International Conference on Networking and Services, ICNS (2013)
8. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
9. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York (2014)
10. Richart, M., Baliosian, J., Serrat, J., Gorricho, J.L.: Resource slicing in virtual wireless networks: A survey. IEEE Trans. Netw. Serv. Manag. **13**(3), 462–476 (2016)
11. Riggio, R., Bradai, A., Harutyunyan, D., Rasheed, T., Ahmed, T.: Scheduling wireless virtual networks functions. IEEE Trans. Netw. Serv. Manag. **13**(2), 240–252 (2016)
12. Sciancalepore, V., Samdanis, K., Costa-Perez, X., Bega, D., Gramaglia, M., Banchs, A.: Mobile traffic forecasting for maximizing 5G network slicing resource utilization. In: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, pp. 1–9. IEEE (2017)
13. Sun, G., Xiong, K., Boateng, G.O., Liu, G., Jiang, W.: Resource slicing and customization in ran with dueling deep q-network. J. Network Comput. Appl. **157**, 102573 (2020)
14. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (2018)
15. Taleb, T., Afolabi, I., Samdanis, K., Yousaf, F.Z.: On multi-domain network slicing orchestration architecture and federated resource control. IEEE Network **33**(5), 242–252 (2019)
16. Van Huynh, N., Hoang, D.T., Nguyen, D.N., Dutkiewicz, E.: Optimal and fast real-time resource slicing with deep dueling neural networks. IEEE J. Sel. Areas Commun. **37**(6), 1455–1470 (2019)
17. Vincenzi, M., Antonopoulos, A., Kartsakli, E., Vardakas, J., Alonso, L., Verikoukis, C.: Multi-tenant slicing for spectrum management on the road to 5G. IEEE Wireles Commun. **24**(5), 118–125 (2017). https://doi.org/10.1109/MWC.2017.1700138
18. Wang, G., Feng, G., Quek, T.Q., Qin, S., Wen, R., Tan, W.: Reconfiguration in network slicing - optimizing the profit and performance. IEEE Trans. Netw. Serv. Manag. **16**(2), 591–605 (2019)
19. Wang, G., Feng, G., Tan, W., Qin, S., Wen, R., Sun, S.: Resource allocation for network slices in 5G with network resource pricing. In: GLOBECOM 2017-2017 IEEE Global Communications Conference, pp. 1–6. IEEE (2017)
20. Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., Freitas, N.: Dueling network architectures for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1995–2003. PMLR (2016)
21. Watkins, C.J., Dayan, P.: Q-learning. Mach. Learn. **8**(3–4), 279–292 (1992)

Chapter 4
AI-Based Performance Enhancement for Multi-Tenant Slicing

4.1 New Business Model for Multi-Tenant Slicing

4.1.1 Resource Sharing Scenarios Among Tenants

In the evolution process of wireless network, the explosive data traffic makes it urgent to improve network capacity, which results in dense deployment of base stations with small coverage, low power, and low cost [4]. The trend of network densification and the pursuit of high data rates prompt network operators to find new cost-efficient solutions to reduce CAPEX/OPEX costs. Network sharing which enables mobile operators to offer services with reduced costs by sharing network infrastructures was explored in the past and partially deployed. The 3GPP Services Working Group SA1 specified five main business scenarios for network sharing [24]:

- **Multiple core networks sharing a common RAN**, where operators share RAN elements, but not the spectrum.
- **Operator collaboration to enhance coverage**, where two or more operators with individual frequency licenses and respective RANs together provide coverage.
- **Sharing coverage in specific regions**, where one operator provides shared coverage and other operators are allowed to use it in this area.
- **Common spectrum sharing**, where the spectrum are shared by a number of operators.
- **Multiple RANs share a common core network**, where multiple RANs belonging to different operators share a common core network.

In the evolution from network sharing to multi-tenancy, vertical industries and OTT providers trend to request network resources with customized capabilities form InPs to provide services. Based on the virtualization mechanisms and software-based capabilities, the notion of network slicing is enhanced for supporting on-

W. Guan, H. Zhang, *Network Slicing for Future Wireless Communication*, Wireless Networks, https://doi.org/10.1007/978-3-031-58229-5_4

demand multi-tenant mobile network. Heterogeneous resources of infrastructure network can be shared by different tenants via network slicing, which offers significant opportunities for OPEX reduction and limits the cost increase in larger coalitions [34]. Some key technical challenges of realizing flexible multi-tenant slicing and dynamic QoS provision are listed here:

- **Self-organizing network function** is responsible for self-configuration, self-optimization, and self-healing of slices. Multi-tenancy brings challenges in automation of the sharing mechanisms and the joint resources optimization.
- **Service exposure function** acts as SLA negotiation intermediary and is responsible for network functionalities exposure. As for multi-tenant slicing, it is challenging to guarantee slice isolation for third parties protection and realize access authorization for MNOs/InPs safeguard.
- **Slice management and orchestration function** support flexible RAN or Core network slicing and dynamic transport network slicing according to SLAs, carrying out VNFs allocation and mobility management. Multi-tenant slicing related challenges lie on satisfying the differentiated requirements of E2E slices, offering efficient management and on-demand orchestration of VNFs belonging to different slices.
- **E2E slicing bargaining function** as a slice auctioneer between the network owners and the third parties performs bargaining SLAs with infrastructure owners, monitoring the allocated slices and trading for dynamic adaptation to variable requirements. In multi-tenant slicing, joint dynamic planning and negotiation of VNFs are challenging, especially for time-critical services.

Enabling efficient sharing of mobile network resources is a key problem underlying multi-tenant slicing. Tenants as slice owners should own the capability of customizing resource allocation within their slices while guaranteeing slice isolation from one another. To the best of our knowledge, there is a vast literature addressing resource allocation problem of network slicing. Existing studies cover the orchestration of core network slicing [3, 18], resource scheduling of RAN slicing [14, 23], and cross-domain implementation of E2E slicing [7, 35]. However, few of existing dynamic resource allocation model focus on the differences in resource preferences of multiple tenants based on perceived congestion at resources. There already has some researches concerning the network slicing game aimed at optimizing the benefits of resource sharing among tenants. Authors in [5] analyze the efficiency and fairness of the resulting resource allocations and provide a dynamic resource sharing mechanism across slices.

With the addition of more and more vertical industries, the number of tenants is increasing, and the types of slicing are gradually increasing. The resource sharing strategy for multi-tenant slicing needs to solve the following problems:

- **The state of the network becomes more complex.** On the one hand, the number of users corresponding to multiple types of services is large and the types vary greatly, and the size, rate and change scale of network flows are different, so resource allocation needs to be adjusted in real time according to time-varying

traffic demands [12]. On the other hand, real-time reconfiguration of network slices is not only the proportional increase or reduction of resources for the VNFs, but also the migration of VNFs [31], which often results in inefficient utilization of resources due to the lack of global cognition of physical resource usage, infrastructure network topology and other information. Therefore, how to obtain accurate network status information in time is essential in the process of slice resource allocation.

- **The demand of slices is hard to be satisfied in time.** With the increase of the number of customized slices, it requires huge computational overhead and complexity to avoid conflict caused by resource competition while ensuring slice isolation. In addition, more and more services require strict E2E delay and bandwidth guarantee, which undoubtedly requires rapid and efficient response to demand changes and optimal resource scheduling strategies [19]. At the same time, the increase of slice types will also lead to more differences in the demand of heterogeneous resources in different domains, which undoubtedly intensifies the decision-making burden of centralized resource management. It is important to perceive the changing trend of service demand and deal with the uncertainty caused by real-time changes efficiently and quickly.
- **Joint optimization of heterogeneous resources becomes difficult.** The participation of vertical industries will lead to a more complex resource allocation scenario, and the game between different stakeholders will lead to more intense resource competition. Resource sharing with interest competition requires reasonable resource pricing and trading strategies, which can be adjusted in time according to dynamic demand changes [2]. Distributed resource management is difficult to adapt to the hierarchical resource allocation system that InPs allocate resources to tenants and tenants allocate resources to users. It is necessary to take into account the global income while ensuring the local equity in the joint optimization of resources.

In order to solve the above problems, as a key technology to realize AI, ML has become an important way to improve the efficiency of network resource management in the field of wireless communication with its advantages in solving complex problems in dynamic environment [29]. Many research institutions have begun to use ML-based methods to realize optimal slice resource allocation, making decisions quickly and in time. It is of great significance for the future development of mobile communication to explore AI-driven slice intelligent management, enhance the ability of automatic network management and resource optimization, and meet the needs of efficient operation in a complex networking environment.

4.1.2 Collaborative Business Model for Multiple Tenants

5G networks have been deployed commercially at the end of 2019 and researches on 6G networks are under way in several countries and organizations [37]. Network

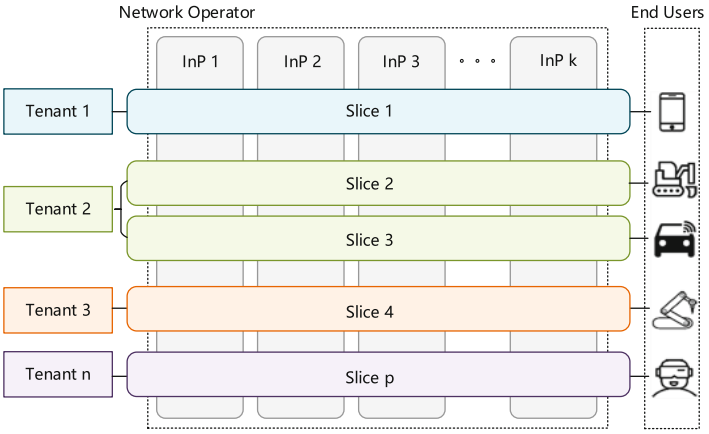


Fig. 4.1 The new collaborative business model of multi-tenant slicing

slicing as a key technology in 5G/6G provides scalability and flexibility in resource allocation, enabling multiple tenants which have different service requirements to share the resources of the 5G/6G network infrastructure. Slicing makes possible the realization of multiple logical networks for multi-tenants on a single shared physical infrastructure [8]. The new collaborative business model of multi-tenant slicing is illustrated in Fig. 4.1, where different communication services are provided to multiple tenants via a number of slices. A large telco corporation acts as network operator which deploys slices over a heterogeneous infrastructure network. Mobile Virtual Network Operators (MVNOs) from vertical industries could leverage on slices provided by the network operator to create their services. End users which are consumers of diverse services could be a mobile phone, cars, or robots.

Based on the above model, a multi-domain orchestration architecture across RAN and transport networks is proposed in [21] to realize network resource sharing toward multiple tenants. Network slices enable tenants which are also perceived as SPs to compete with each other using the same infrastructure. Many studies have focused on the problem of competitive multi-tenant cross-slice resource orchestration [5, 6]. For multi-tenant slicing, efficiently and holistically resource management methods are required while the isolation among slices needs to be guaranteed. On the one hand, dealing with the uncertain traffic condition of slices in dynamic environment, such as flow arrival/departure, means that slice reconfiguration should be executed in a timely manner. Making decision of resource allocation strongly depends on the current status of slices as well as their predicted demands. On the other hand, different tenants use slice instances to provide services with conflicting resource requirements, resulting in that providing isolation becomes hardly and costly.

Given that the deployment of slices include the mapping of VNFs and the virtual links between VNFs, achieving slice isolation is often accompanied by the cost of reducing multiplexing gain and inefficient resource utilization. To ensure strict QoS,

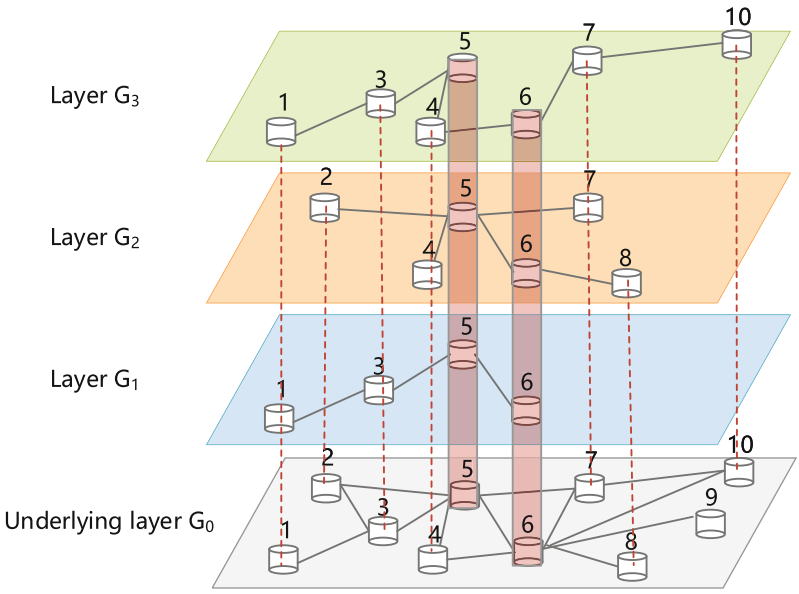


Fig. 4.2 The schematic illustration of resource competition among layers

tenants as service providers should avoid the degradation of traffic performance of slices when the number of colocated VNFs is high. According to empirical studies, VNFs deployed as a VM on the same server co-exist with many other VMs, which will affect network performance when faced with bulk traffic load [9]. Especially, when the number of VMs increases to a certain extent, the stringent end-to-end latency of slices will be damaged. Hence, a comprehensive performance measurement and analysis is required to avoid the effect of resource competition.

In order to analyze the impact of resource competition on traffic performance of network slices, a multilayer network model is used here to represent the deployment of multi-tenant slices in a large infrastructure network. As shown in Fig. 4.2, the multilayer network model consists of an underlying infrastructure network G_0 with N nodes and M upper layers, for each $m \in M$, slice G_m as the upper layer consists of N_m nodes, $1 \leq m \leq M$, $N_m \leq N$ and $N_m = N \cdot P_m$. The nodes of the underlying layer G_0 denote the physical standardized servers of the infrastructure network and the nodes of layers G_m denote the VNFs of slices. P_m is the nodal coverage of slice G_m, which represents not only the density of nodes but also the overlap of nodes.

Traffic flows belonging to slice G_m occur on edges wholly within the specific layer, which can only be transferred within G_m. Node i of G_0 can appear in many different layers, which means several slices will share the resources of the same physical node of G_0 through the interconnections, e.g., nodes 5 and 6 in Fig. 4.2. Layer G_1, layer G_2, and layer G_3 are three different slices deployed in the underlying infrastructure network G_0. Note that this topology is used to illustrate the generation process, and it is not involved in the dynamic analysis. In this schematic

example, $N = 10$ is the number of nodes in the underlying layer; $M = 3$ is the number of slices. Nodes 5 and 6 exist in these three slices, which means that these three slices are competing for physical resources of node 5 and 6.

This multilayer network model provides a clear global perspective for service providers to analyze the resource competition among slices, guiding the utilization of resources to satisfy the requirements of slices.

4.2 Traffic Performance Analysis of Multiple Isolated Slices

4.2.1 Traffic Model of Multiple Slice

For accurate and convenient evaluation of traffic performance, the topology generation process of the underlying infrastructure network and slices are introduced. Since there is a trend of cloudifying the mobile network infrastructure and building federated cloud infrastructures [30], service providers deploy more regional DCs distributed geographically. VNFs of E2E slices are placed as VMs which are run on the physical DCs. DC which is more important in service delivery has a much higher probability to be connected by the new added DC. Considering that a forthcoming node has the tendency to connect itself to the nodes with large degrees in the growth of communication network, the algorithm of *BA* scale-free networks is adopted to generate the topology of the infrastructure network and slices. As one of the basic network structures of practical relevance, adopting BA growth model could intuitively show the feature that very few DCs have much higher utilization than others in the infrastructure network.

The underlying network G_0 is generated according to the *BA* generating algorithm, which are introduced in Chap. 2. After generating the underlying infrastructure network, each slice is generated via the following procedure:

1. Given the degree $d(i)$ of node i in G_0, its selection fitness ϕ_i is given by

$$\phi_i = [d(i)]^{\varepsilon}, \varepsilon \in (0, 1), \tag{4.1}$$

   where ε is a parameter denoting the degree of preferential picking.
2. Randomly select a node in G according to the ϕ_i, and the possibility of selecting node i is $p(i)$,

$$p(i) = \frac{\phi_i}{\sum_{k=1}^{N} \phi_k}. \tag{4.2}$$

3. Randomly select its $P_m \cdot d(i)$ neighbors.
4. For each selected neighbor, repeat step 3 until the total number of selected nodes reaches N_m.

Besides, a degree-based resource allocation scheme is provided and two typical measures is defined to quantify the traffic performance. In order to simplify the implementation, here the resources are represented by the buffer size of nodes. Under the degree-based resource allocation scheme, the buffer size $B(i)$ for slice G_m is allocated based on the total number of nodes N_m in this layer. In other words, the scale of the slices decides the total amount of resources they owned. Hence,

$$B_m = \frac{(N_m)^\alpha}{\sum_{m=1}^{M} (N_m)^\alpha} \cdot B_{all}, \tag{4.3}$$

where B_m denotes the total resources in slice G_m, that is the sum of buffer sizes of nodes in slice G_m, and $\alpha \in [0, 1]$ is a parameter which morphs the resource of slices from a uniform distribution to one dependent on the number of nodes in the slices. Since the nodes with higher degree play an important role in the transmission of traffic data packet, the buffer size of node i in slice G_m can be allocated according to its intra-layer degree $k_m(i)$. It can be denoted as

$$B_m(i) = \frac{(k_m(i))^\beta}{\sum_{i=1}^{N_m} (k_m(i))^\beta} \cdot B_m, \tag{4.4}$$

where $k_m(i)$ is the degree of node i in slice G_m. $B_m(i) = 0$ when node i does not exist in slice G_m, and $\beta \in [0, 1]$ is a second strength parameter which is the same as α.

The traffic performance of slices is evaluated by the situation of data communication in these slices. Wireless network is a generic type of communication networks, but beyond that the E2E slices regarded as virtual E2E wireless networks cross three different domains, RAN, transport network, and core network. The data or information in slices can be presented as packets and transmitted through links between pairs of nodes under specific routing algorithm.

In order to describe the dynamics of traffic data packets, the principles of traffic generation are provided. There are two types of nodes. *Routers* only store and forward packets; *hosts* can also generate packets. The density of hosts $\rho \in (0, 1)$ is the ratio of the number of hosts to the total number of nodes. Here, we set $\rho = 0.4$ and randomly select hosts in each layer. The packets are operated as follows:

1. Packet generation: at each time step, new packets are generated on the hosts. The generation rate is λ, which means that the number of generated packets in each layer is λ.
2. Packet transmission: each previously generated packet can move freely along the shortest path toward its destination. At each time step, δ packets are forwarded to the target neighbor of the node by one step. δ is the transmission rate, we set $\delta = 1$ in our model for simple simulations.

3. Transmission path changing: if the number of packets reaching target neighbor i is larger than its buffer size $B_m(i)$, the transmission path will be changed to a randomly chosen neighbor which has some buffer space, provided that such a neighbor exists. If all the neighboring nodes have full buffers, then the packet will remain at the original node.
4. Packet released: packets will be released once they arrived their destination.

The transmission protocol of packets within layers requires that packets are generated in hosts of the layer with destination addresses and are transferred by the routers one hop at a time toward their destinations by the shortest path. Each node in each layer has a buffer for storing packets and the buffer size is the result of allocating the limited capacities in the underlying network. Here the total buffer size of a node is considered as the resource of the underlying network, which are sliced by different layers in a particular way. For a fair comparison, the total buffer size of all nodes are kept the same in all simulations. With the same total resources, the traffic performance is analyzed under a degree-based resource allocation.

To quantify the traffic performance of slices based on the multilayer network model, two measures are defined: (1) The ratio of arrived packets R, which is the ratio of the numbers of successfully arrived packets to the total generated packets. (2) The average travel time T of packets in a particular slice is the average latency between sending and receiving of packets. The ratio of arrived packets can be calculated with

$$R_m = \frac{Num^{G_m}}{\lambda \cdot t_m}, \tag{4.5}$$

where t_m is the length of the time step. Since λ is the number of generated packets in each time step, $\lambda \cdot t_m$ is the total number of generated packets. The average travel time T of layer G_m is defined as

$$T_m = \frac{1}{Num^{G_m}} \cdot \sum_{p=1}^{Num^{G_m}} \left(t_p^{OUT} - t_p^{IN} \right), \tag{4.6}$$

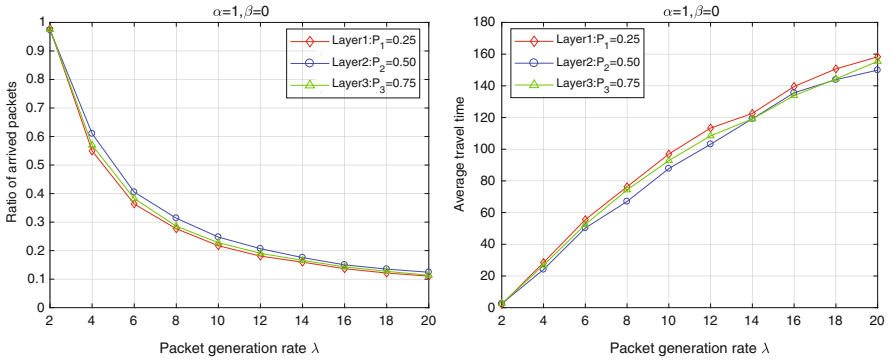
where t_p^{OUT} and t_p^{IN} are the time at which packet p enters the network and arrives at its destination, respectively. Num^{G_m} is the total number of successfully arrived packets.

4.2.2 Performance Analysis of Slice Traffic

In this section, we present the numerical experiments to analyze the traffic performance of slices using MATHEMATICA 11.0 and MATLAB 2015b. Each realization runs for 200 time steps and simulation under the same settings has been operated 100 times independently to obtain the average. Besides, with the increasing of the

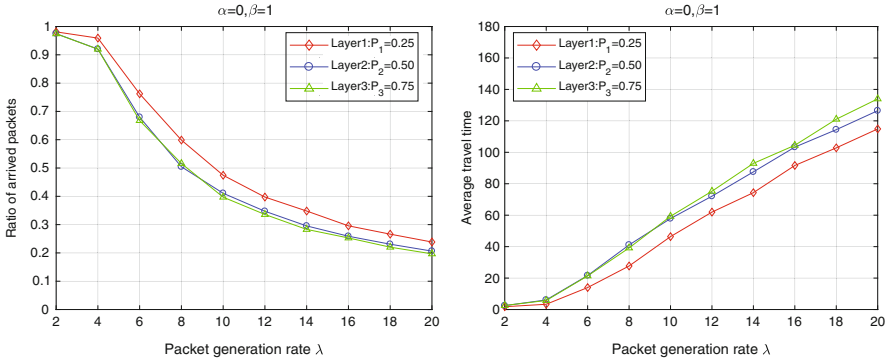
packets generation rate, there are numerous packets needed to be transmitted in slices and the transmission state of each packet needs to be recorded and updated at every time step. Hence, the time required to get the results increases dramatically as the multilayer network model scales up. Most of our simulations are for the multilayer network which has three slices sharing the resources of the underlying infrastructure network. It is not only because the time and memory space of a laptop are limited but also the conclusions will remain the same when the number of slices is higher.

Figures 4.3 and 4.4 show the behavior of R and T for different nodal coverage values and resource allocations. Here the numbers of nodes vary across the layers, which means that three different slices have different nodal coverage P_m. The total number of physical nodes in the underlying infrastructure network is 20, layer 1 has 5 nodes, layer 2 has 10 nodes, and layer 3 has 15 nodes, thus $P_1 = 0.25$, $P_2 = 0.5$, and $P_3 = 0.75$. Assuming that the total capacity of the underlying infrastructure network is 60, $B_{all} = 60$. In order to eliminate the randomness and improve the



(a) (b)

Fig. 4.3 The traffic performance of differentiated slices when $\alpha = 1$, $\beta = 0$



(a) (b)

Fig. 4.4 The traffic performance of differentiated slices when $\alpha = 0$, $\beta = 1$

accuracy, the simulations have been operated 100 times independently and the average has been taken. Two typical cases of resource allocation are observed, $\alpha = 1, \beta = 0$ and $\alpha = 0, \beta = 1$.

The parameter choice $\alpha = 1, \beta = 0$ means that the slice with greater density of nodes has more resources which is allocated equally and the buffer size of each node is the same value. By contrast, the choice $\alpha = 0, \beta = 1$ provides the same resources to each slice and these resources are allocated to a slice's nodes according to the degrees of these nodes. In this case, the nodes with higher degree in the slice with lower nodal coverage have more resources. In Fig. 4.3, the ratio of arrived packets for the lowest packet generation rate ($\lambda = 2$) is closing to 1, and the average travel time is 0, implying that all the packets can be transferred efficiently in each slice. As the packet generation rate is increased, each slice suffers from an increasing shortage of resources. The numbers of successfully arrived packets for each slice decreases rapidly and packets spend more time on travel. However, the rate of descent varies from slice to slice and the difference between slices also varies according to the differing resource allocations.

As shown in Fig. 4.3a, b, R and T of three slices are close but layer G_2 has higher R and lower T. R of layer G_1 is lowest and T is highest while layer G_3 is somewhere between. The main reason is that the total resource of layer G_1 is least and the average path length of packet travel in layer G_3 is longest. Compared with layer G_1 and G_3, layer G_2 performs better in terms of R and T. It can be concluded that the impact of nodal coverage P_m is smaller than resource allocation α and β when all nodes have the same buffer size. In Fig. 4.4a, b, layer G_1 has highest R and lowest T obviously while R and T of layer G_2 are slightly better than layer G_3. In this case, nodes in layer G_1 are allocated more resource than the other two layers and important nodes in all layers have larger buffer size.

Comparing these two typical cases, it can be observed that the better resource allocation method can improve the overall traffic performance in terms of the ratio of arrived packets and the average travel time. Since α determines the total resources of slices and β determines the effect of nodes degree on resource allocation, how to allocate the buffer size to nodes in different slices play a more important role.

In order to understand the effect of resource allocation, the traffic behavior is investigated by varying α and β. Figure 4.5 shows the average travel time of packets in different slices versus various α and β. In Fig. 4.5a–i show the average travel time of layer 1,2,3, respectively, versus the resource allocation parameters α and β. $\lambda = 5$ and other simulation parameters are the same as in Fig. 4.3. In order to analyze the impact of resource allocation further, the underlying infrastructure network is changed in two different ways: (1) using the same generation algorithm to generate a different set of edge, (2) replacing the generation algorithm of *BA* scale-free network with the generation algorithm of *Erdös-Renyi* (*ER*) random network. For the *ER* random network, the probability of connecting each pair of nodes by an edge is set to 0.2, and the number of edges is almost equal to *BA* configuration. *ER* random network has the same number of nodes as the *BA* scale-free network.

In Fig. 4.5a, the average travel time T increases obviously with α but it looks pretty much the same for various β. This is because the growth of α will lead to

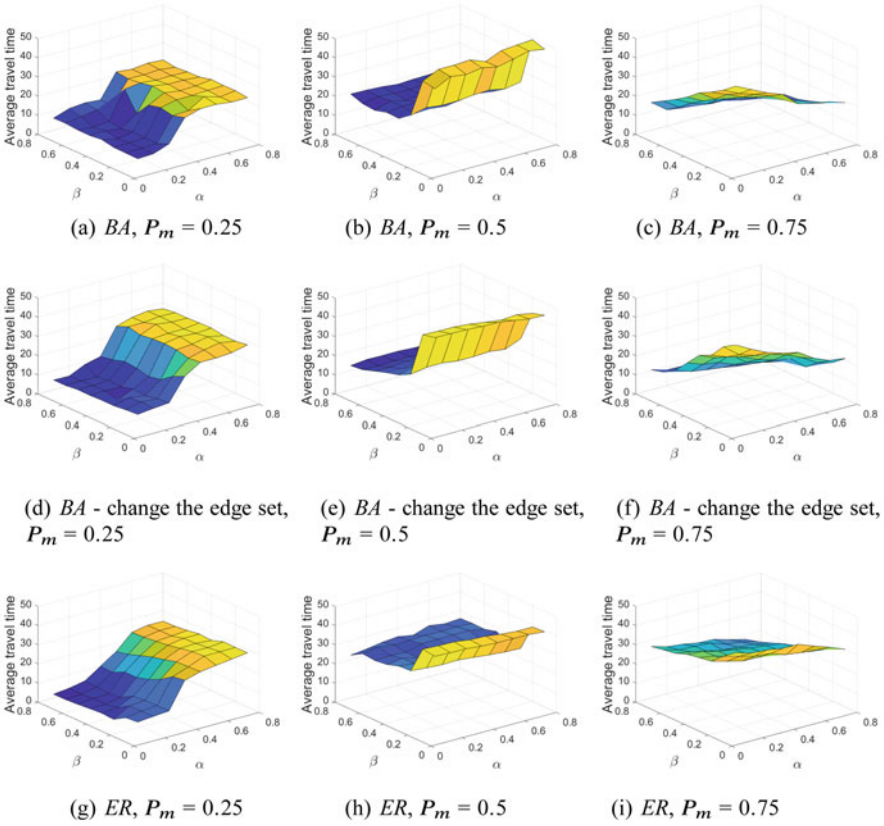


Fig. 4.5 The effect of resource allocation parameters

the reduction of total resources for layer G_1, and it takes more time to transfer the packets. In contrast, as a result of small nodal coverage, the increase in β does not have significant impacts. For layer G_2, changing α and β has an opposite impact. As shown in the Fig. 4.5b, reducing the value of β can effectively reduce the transmission time. It means that packets can be transmitted faster with the degree-based resource allocation method. Since $B_2 = \frac{1}{3}B_{all}$ for $\alpha = 1$ and $\alpha = 0$, changing α from 0 to 1 has little effect.

The results of increasing the total resources of layer G_3 and distributing the resources to nodes based on their degrees are shown in Fig. 4.5c. T for layer G_3 will be reduced when either α is increased or β is increased. Hence, it can be found that these two influencing factors have different impact on traffic performance for layers with different nodal coverage. The results are effected by the nodal coverage of slices and the number of slices. The result of the first change is shown in Fig. 4.5d–f. This shows that the effect of changing α and β is the same when the generation algorithm does not change.

As shown in the Fig. 4.5g–i, the variation tendency of average travel time is basically the same, which means that the impact of changing α and β still remains unchanged. However, the average travel time is much higher than the *BA* algorithm when β is higher for layer 2 ($P_m = 0.5$) and layer 3 ($P_m = 0.75$). Since the scale of layer 1 ($P_m = 0.25$) is quite small, the average transmission path length of *ER* random network is shorter than *BA* scale-free network. Hence, in Fig. 4.5g, the average travel time is small when $\alpha \leq 0.4, \beta \geq 0.4$. It can be observed that the degree-based resource allocation method is more suitable when the underlying infrastructure network is a *BA* scale-free network. The reason is that the degree distribution of scale-free network is a power law, and a small number of nodes have high degree. The existence of these nodes can reduce the average transmission path length between any two nodes. Providing more resources to these node will shorten the average travel time efficiently.

The level of congestion found at the node i in layer G_m is denoted as

$$L_i^{G_m}(t) = \frac{q_i^{G_m}(t)}{B_m(i) - q_i^{G_m}(t)} \tag{4.7}$$

$q_i^{G_m}(t)$ represents the number of packets arrived in node i at time t, and $B_m(i)$ denotes the capacity of storing packets at node i. In order to analyze the effect of changing the nodal coverage and the number of slices, the ratio of congested nodes is counted when the time step is 200 and different slices have the same nodal coverage. It can be defined that node i in layer G_m is a congested node when $L_i^{G_m}(200) \geq 1$.

Figure 4.6 shows the ratio of congested nodes versus the nodal coverage of slices and the number of slices. The underlying network is a *BA* scale-free network. $\alpha = 0.5, \beta = 0.5, \lambda = 5$ and other simulation parameters are the same as in Fig. 4.3. Figure 4.6a is the result of how the congestion state of slices vary with M when the nodal coverage of slices (P_m) is fixed. The ratio of congested nodes exhibits a faster

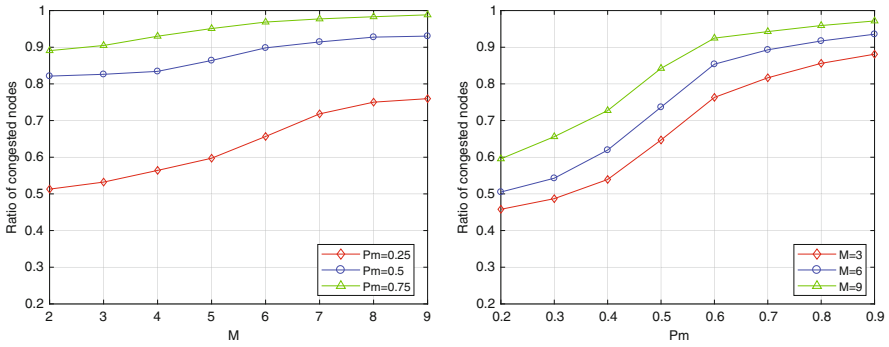


Fig. 4.6 The ratio of congested nodes

increase when P_m is 0.25 than it is 0.75 and 1. Most of nodes in layer G_2 and layer G_3 are already congested when M is small. It means that increasing the number of slices has a more significant impact when the value P_m is less.

Figure 4.6b shows that the ratio of congested nodes increases much faster with the number of slices when M is 3, 6, 9. By comparing these two figures, it can be observed that the resilience against congestion in the multilayer network model is more sensitive to the increase of P_m than M. A higher value of M means that the total resources for each slice is decreasing. And a higher P_m means that it is likely for more nodes in different slices to share the resources of common nodes in the underlying infrastructure network. Hence, a high value of P_m will cause more severe congestion and an arbitrarily small change in P_m can drive the multilayer network into a congested state. Thus, P_m has a critical impact at these values.

Network slicing is modeled by the multilayer model where traffic packets are generated and transmitted in the slices and the buffer sizes of nodes in the slices are perceived as the resources allocated according to the degree of nodes. The impact of packet generation rate λ, the resource allocation parameters α and β, the number of slices M and the nodal coverage P_m on the ratio of arrived packets R and the average travel time T are analyzed. The results show that a reasonable resource allocation method can improve the traffic performance, and the degree-based method is suitable when the underlying infrastructure network is a scale-free network. Moreover, adjusting the resource allocation parameters has different effects on slices because of different nodal coverage P_m. However, the traffic performance on slices can be improved by increasing the total resources of slices and allocating more resources to nodes with higher degree. After obtaining the variation of the ratio of congested nodes when changing M and P_m, it is found that increasing P_m has a deeper impact on triggering the congestion than increasing M.

4.3 Inter-Slice Resource Sharing and Competition

4.3.1 Control Strategy for Avoiding Resource Competition

The analysis in above section shows that a minor adjustment in the resource allocation parameters α or β can result in an increase in average travel time for slices. Increasing the total resources of a slice or allocating more resources to nodes of higher degree can reduce the average travel time for packets. It can be concluded that the hub nodes in the multilayer network model, as the top nodes ranked by degree, play a crucial role in causing congestion. Hence, a practical strategy of preventing the congestion is to selectively enhance the buffer capacity of a few hub nodes in the underlying infrastructure network. In practice, this strategy means that InPs increase computing resources of a few more important DCs to maximize global resource efficiency with capacity expansion methods. With the real-time status of slices and economic benefit analysis, the revenue of maintaining high service quality could be obtained by sacrificing small reinvestment cost.

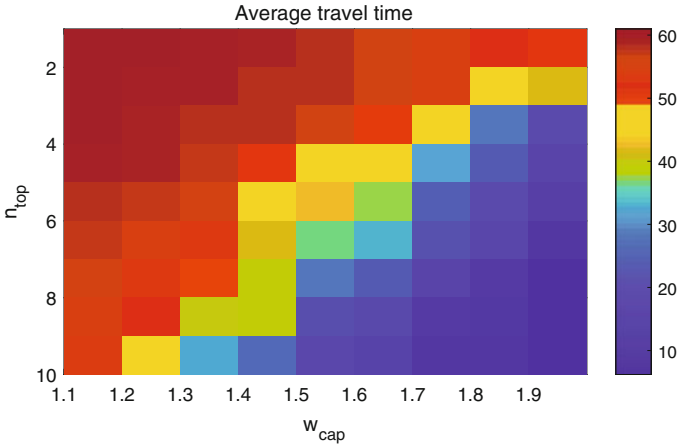


Fig. 4.7 Average travel time of changing the capacity of hubs

Giving more resources to a small set of hub nodes is effective but costly. Hence, the number of these top nodes n_{top} and the increment of capacity need to be reasonable in order to achieve a trade-off. The buffer capacity of these top nodes can be increased by multiplying a factor $w_{cap} \geq 1$. For example, the congestion can be prevented by increasing the buffer size of the top 2 hubs by 1.5 times when the total number of nodes in the underlying infrastructure network is 20 (2 out of 20, 10%). Comparing the total capacity of the underlying infrastructure network, the enhanced buffer size is insignificant but it can reduce the average travel time drastically.

Figure 4.7 shows the change of the average travel time for a slice when the buffer sizes of the top n_{top} nodes are multiplied by the factor w_{cap}. After finding n_{top} nodes, the newly increased buffer size is allocated in the same way as before with the same α and β. Here we set $\alpha = 0, \beta = 1$, and $\lambda = 10$. There are 3 slices $M = 3$ and they have the same nodal coverage $P_m = 0.5\,(m = 1, 2, 3)$. There exists a region, the blue area, in the parameter space that represents a completely free state of traffic flows, with a clear boundary separating this region from the congestion regions, the red area. As shown in Fig. 4.7, neither too small values of n_{top} nor small values of w_{cap} can efficiently reduce the travel time. It again shows that providing more resources to a select set of hubs can avoid the congestion state efficiently.

According to Fig. 4.6, a small increase in the parameter P_m can lead to a substantial increase of the congested nodes suggesting that the parameter P_m can deliver a critical change in network congestion. A straightforward control strategy is to reduce the value of P_m of these slices. Since the congestion can be triggered by these nodes shared by many slices, reducing the overlap between layers can suppress the congestion efficiently. However, this method will not be effective when P_m is

high enough ($P_m \geq 0.6$ in Fig. 4.6b). Also, this method is not practical in real communication networks because the structure of slices are predetermined.

However, the scale of the underlying infrastructure network can be increased since many real multilayer networks are evolved by adding the nodes and connections, such as a social network, the transportation network, and the urban infrastructure network. As for the cloud infrastructure of the wireless network, increasing its scale could be realized by cooperating with other infrastructure networks or deploying more DCs to form a federated environment. Since the underlying infrastructure network is a *BA* scale-free network, the scales of the underlying infrastructure network can be increased by increasing the total number of nodes with the generation algorithm in Sec. II. Then, the overlap between slices can be reduced while the numbers of nodes in these slices are kept. This action results in an increase in the ratio of arrived packets. However, this strategy has a limit that the cost of increasing the scale of the underlying infrastructure network outweigh the costs of increasing the buffer capacity of hubs.

As shown in the Fig. 4.8, the ratio of arrived packets is increased rapidly when the underlying infrastructure network is scaled up. In this figure, the total number of nodes is increased from 20 to 60 step by adding 10 new nodes. Also, the total capacity is increased from 60 to 180 in order to ensure the average buffer size of each node is 3. Here we set $\alpha = 0$, $\beta = 1$, and $\lambda = 10$. There are 3 layers $M = 3$ and the numbers of nodes for layers are 5,10,15. Compared to layer 2 and 3, the ratio of arrived packets for layer 1 is doubled by adding 10 nodes. It means that this strategy can improve the performance dramatically for slices with relatively few nodes. Furthermore, for all three slices, the ratio of arrived packets increases slowly

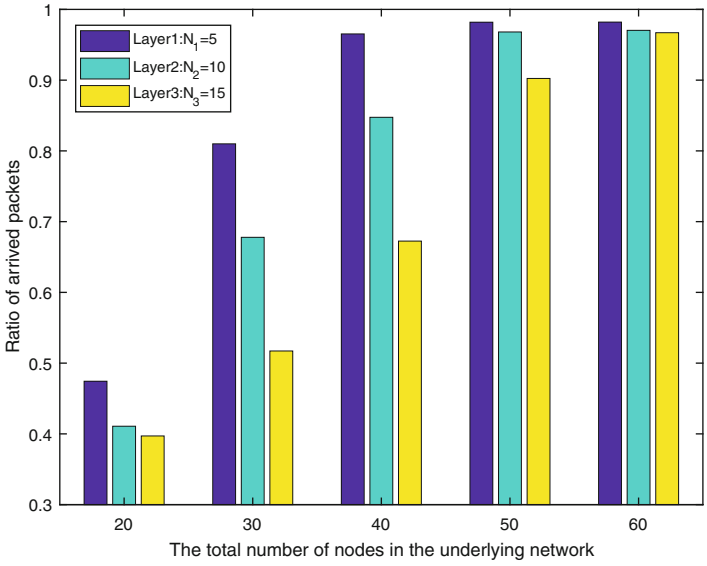


Fig. 4.8 Ratio of arrived packets when changing the scale of the underlying network

when it is larger than 0.9. Hence, another limit of this strategy is that the increase in the ratio of arrived packets will stop when the underlying infrastructure network reaches its overall capacity (i.e., $N = 60$).

Based on the performance analysis of slice traffic, two different control strategies are proposed to improve the traffic performance, increasing the buffer capacity of hubs and increasing the scale of the underlying infrastructure network. The former reduces the average travel time by giving more resource to a small set of hub nodes. The latter improves the ratio of arrived packets by adding more physical servers to the underlying infrastructure network. In general, these two strategies reduce the possibility of service quality deterioration caused by resource competition from two aspects, respectively. The first strategy pays more attention to the impact of the several hub nodes on service quality, thus it costs less by expanding the capacity of these hub nodes. Oppositely, the second strategy focuses on the significance of overall available computing resources for improving service quality, thus the performance improvement is more remarkable. Furthermore, the effectiveness and limitations of these two strategies are estimated, which provide a good understanding of controlling the congestion and enhancing the resilience of NS in wireless networks. In addition, the analysis results could be used as a benchmark or reference to the analysis of other multilayer systems.

Considering that the scales of many real world multilayer networked systems are ten times or a hundred times more than this model, it requires multiple processors with good performance to work at the same time. Besides, in order to reduce the complexity and time of processing real data, the underlying infrastructure network and the slices can be divided into multiple different parts according to certain methods. In future research, it will be better to find a suitable division method with a view to realizing the parallel processing of real data. Since minimizing transmission delay of data traffic can meet the stringent delay requirements of services and increase the benefits of service providers, the shortest path routing algorithm is adopted in the traffic generation model in this paper. In order to further analyze the impact of routing algorithms on traffic performance of slices, various differentiated routing algorithms should be selected for different types of services in the future. In addition, joint allocation of multiple types of resources will also be taken into account, including the capacity and delay of virtual links.

4.3.2 Application of AI Techniques in Multi-Tenant Slicing

The wireless communication system has been continuously evolving to provide ultra-fast speed, greater capacity, and ultra-low latency, supporting new applications. With the proliferation of smart devices, the expansion of network scale and the diversification of services, the mathematical formulations of existing algorithms become more complex and they are incapable of solving the problems in dynamic network environment.

As an important enabling technology for AI, ML has been successfully applied in many areas, including computer vision, medical diagnosis, search engines, and speech recognition [15]. ML techniques which can be generally classified as supervised learning, unsupervised learning and RL gives computers the ability to learn without being explicitly programmed. In recent years, many efforts have been made to use ML in wireless communication, including resource management, networking and mobility management, and so on. The motivation of authors in current researches to adopt ML-based methods includes many aspects [29], for example, researches use ML to solve wireless problems with low complexity [1], model-free RL can help the resource scheduler make optimized decisions without a full knowledge of network information [10]. Moreover, signaling overhead can be reduced and better performance can be achieved with ML-based methods since they can learn an optimal decision with partial network information using deep neural network [22].

There already have many some literatures investigated the application of ML in network slicing. Authors in [20] use RL to allocate resources for RAN slices, obtaining better performance than the traditional optimization algorithm in the perspective of optimizing the benefits of InPs. [27] analyzes the advantages of using DRL algorithm to solve problems such as RAN slicing resource allocation, automatic selection of radio access technology, and mobile edge caching. Authors in [25] study the traffic forecasting of network slicing based on past information and admission control of slice requests from different tenants. Further, they proposed an online RL based algorithm for multi-class slice scheduling to improve resource utilization in [26]. Although these researches achieve optimal resource partitioning for slices belonging to different tenants with distinct requirements, it is still difficult to cope with the dynamic resource demand changes.

In multi-tenant slicing, each tenant acts selfishly in order to compete with other tenants for limited resources. However, most of the optimization goals of existing resource allocation algorithms are short-term gains for a single tenant. In order to deal with the uncertainty of resource demand changes, authors in [33] use deep reinforcement learning to implement real-time fast resource allocation for multi-type slices. The multi-type slices belonging to different tenants have great differences in the heterogeneous resource requirements of the infrastructure network, and this difference changes dynamically with time, which will undoubtedly lead to the difficulty in solving the joint optimization problem of heterogeneous resources. [17] model the slicing resource management problem between multiple tenants with competing relationships as a stochastic game and propose an online solution based on DRL to approximate the optimal solution.

As one of most important research directions of ML, RL is particularly suitable for decision-making in resource management of multi-tenant slicing. By combining deep learning introduced as a breakthrough technology with RL, both of the learning speed and performance are improved. Consequently, DRL has become an emerging tool to effectively address various problems in the areas of communications and networking [13]. The learning process of DRL is introduced in the Chap. 1 and the DRL-based slice orchestration is proposed in Chap. 3. Noticed that DRL algorithms

consist of value-based methods and policy-based methods. Value-based DRL focus on estimating the value of different states or state-action pairs while policy-based DRL focus on directly learning the optimal policy–the strategy that maps states to actions to maximize cumulative rewards. Deep Q-Learning (DQL) as a value-based method is mostly used for the DRL related works in wireless communication.

Due to the cross-domain deployment of slices, avoiding the damage of resource competition among different tenants in slice performance becomes challenging. The conventional DRL algorithms, such as Double DQL, Dueling DQL, are gradually unable to deal with the challenges brought by resource diversification, demand dynamics, and differentiation. Some newest DRL algorithms, such as Asynchronous Advantage Actor-Critic (A3C) DRL [16], deep deterministic policy gradient (DDPG) [11], and Multi-Agent DRL (MADRL) [36] are adopted in dynamic slicing.

A3C includes two separate neural networks, actor network which learns the optimal policy and critic network which estimates the value function. The architecture of A3C algorithm is shown in Fig. 4.9, combining the strengths of both actor-critic and advantage learning methods which focus on the relative value of actions rather than their absolute value. In A3C architecture, there are multiple parallel agents to explore the environment and learn the optimal policy, achieving faster learning and better performance. Global network as a public neural network model includes the functions of actor network and critic network. There are N worker below, and each worker has the same network structure as the public neural network. Each agent will interact with the environment independently to obtain empirical data. These parallel agents do not interfere with each other and run independently. After each agent interacts with its environment to obtain certain data, it calculates the gradient of the neural network loss function in its own thread. These gradients do not update the neural network in its own thread but update the public neural network. Compared with the DQL algorithm which uses a single agent and a single environment, the convergence speed of A3C is faster and its robustness is stronger.

Considering that the DQL algorithm works only for discrete action spaces and discretizing continuous action space with many dimensions suffers from curse of dimensionality, DDPG as a model-free off-policy actor-critic algorithm is introduced. The DDPG architecture is shown in the Fig. 4.10, where the actor outputs a deterministic action rather than a probability distribution of the action. The critic evaluates the Q value of the state-action pair, not the V value. DDPG is suitable for solving the problem of continuous action space, but it can also solve the problem of discrete action space and discrete continuous mixed action space.

The core of RL is trial and error, in which the agent iteratively optimizes based on the feedback obtained by interacting with the environment. When there are multiple agents interacting with the environment at the same time, the entire system becomes a multi-agent system. The architecture of MADRL algorithm is shown in the Fig. 4.11. Each agent is still following the goal of RL, which is to maximize the cumulative reward that can be obtained, and the change in the global state of the environment is related to the joint actions of all agents. When these agents interact with the environment and one another, we can observe them collaborate,

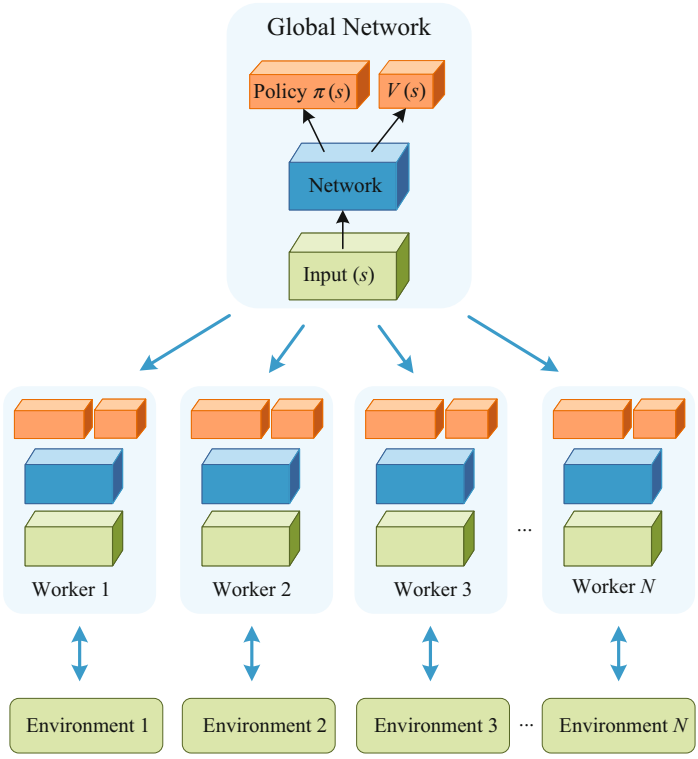


Fig. 4.9 The architecture of A3C algorithm

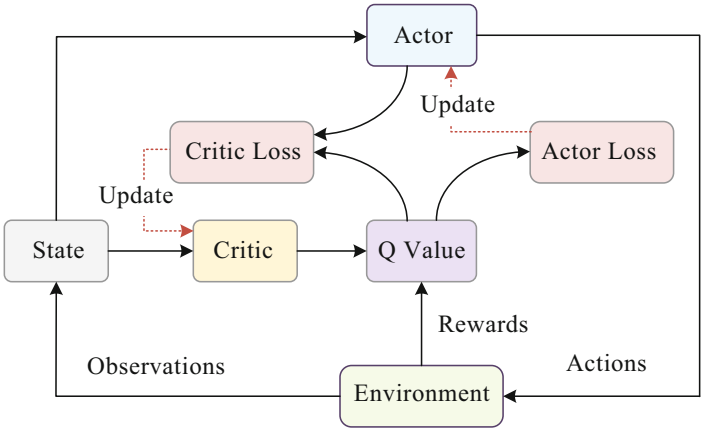


Fig. 4.10 The architecture of DDPG algorithm

coordinate, compete, or collectively learn to accomplish a particular task. The reward function of multiple agents is complex and the relationship between agents can be perceived as a random game. When the reward function of each agent is

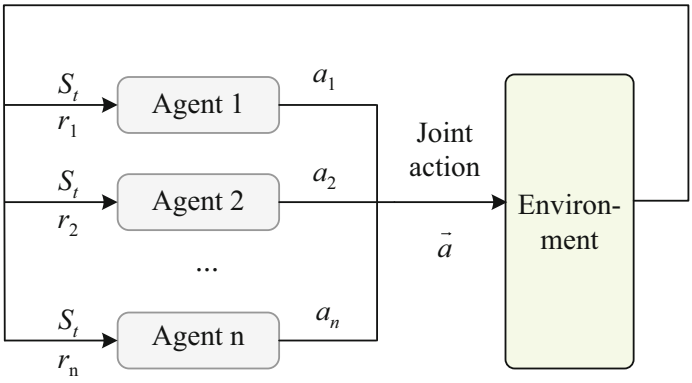


Fig. 4.11 The architecture of MADRL algorithm

consistent, the relationship between agents is cooperative, and when the reward function is opposite, the relationship between agents is competitive. There are also strategies that neither compete nor cooperate, that is, mixed strategies.

Compared with traditional DRL algorithms, these new algorithms are more adaptable to solving complex problems and can learn the optimal decision faster with better performance. Authors in [32] use A3C in jointly optimization of the densities of deployment and resource allocation for RAN slicing relying on control-plane and user-Plane separation. Authors in [28] use MADRL to jointly solve the problems of network slicing and slice admission control. How to use these existing newest DRL algorithms and how to develop new DRL algorithms are essential to achieve automated and intelligent resource management in multi-tenant slicing.

Acknowledgments If you want to include acknowledgments of assistance and the like at the end of an individual chapter please use the `acknowledgement` environment—it will automatically render Springer's preferred layout.

References

1. Ahmed, K.I., Tabassum, H., Hossain, E.: Deep learning for radio resource allocation in multi-cell networks. IEEE Network **33**(6), 188–195 (2019). https://doi.org/10.1109/MNET.2019.1900029
2. Akgul, O.U., Malanchini, I., Capone, A.: Dynamic resource trading in sliced mobile networks. IEEE Trans. Netw. Serv. Manage. **16**(1), 220–233 (2019). https://doi.org/10.1109/TNSM.2019.2893126
3. Bagaa, M., Taleb, T., Laghrissi, A., Ksentini, A., Flinck, H.: Coalitional game for the creation of efficient virtual core network slices in 5G mobile systems. IEEE J. Sel. Areas Commun. **36**(3), 469–484 (2018). https://doi.org/10.1109/JSAC.2018.2815398
4. Bhushan, N., Li, J., Malladi, D., Gilmore, R., Brenner, D., Damnjanovic, A., Sukhavasi, R., Patel, C., Geirhofer, S.: Network densification: the dominant theme for wireless evolution into 5G. IEEE Commun. Mag. **52**(2), 82–89 (2014)

5. Caballero, P., Banchs, A., De Veciana, G., Costa-Pérez, X.: Network slicing games: enabling customization in multi-tenant mobile networks. IEEE/ACM Trans. Networking **27**(2), 662–675 (2019). https://doi.org/10.1109/TNET.2019.2895378
6. Chen, X., Zhao, Z., Wu, C., Bennis, M., Liu, H., Ji, Y., Zhang, H.: Multi-tenant cross-slice resource orchestration: a deep reinforcement learning approach. IEEE J. Sel. Areas Commun. **37**(10), 2377–2392 (2019)
7. Chien, H.T., Lin, Y.D., Lai, C.L., Wang, C.T.: End-to-end slicing as a service with computing and communication resource allocation for multi-tenant 5G systems. IEEE Wireless Commun. **26**(5), 104–112 (2019). https://doi.org/10.1109/MWC.2019.1800466
8. Chien, H.T., Lin, Y.D., Lai, C.L., Wang, C.T.: End-to-end slicing with optimized communication and computing resource allocation in multi-tenant 5G systems. IEEE Trans. Veh. Technol. **69**(2), 2079–2091 (2020)
9. Feng, X., Lu, Z., Wang, L., Guan, W.: A delay-aware deployment policy for end-to-end 5G network slicing. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC), pp. 1–6 (2019). https://doi.org/10.1109/ICC.2019.8761633
10. Li, R., Zhao, Z., Chen, X., Palicot, J., Zhang, H.: Tact: A transfer actor-critic learning framework for energy saving in cellular radio access networks. IEEE Trans. Wireless Commun. **13**(4), 2000–2011 (2014). https://doi.org/10.1109/TWC.2014.022014.130840
11. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. Preprint. arXiv:1509.02971 (2015)
12. Luo, Z., Wu, C., Li, Z., Zhou, W.: Scaling geo-distributed network function chains: a prediction and learning framework. IEEE J. Sel. Areas Commun. **37**(8), 1838–1850 (2019). https://doi.org/10.1109/JSAC.2019.2927068
13. Luong, N.C., Hoang, D.T., Gong, S., Niyato, D., Wang, P., Liang, Y.C., Kim, D.I.: Applications of deep reinforcement learning in communications and networking: a survey. IEEE Commun. Surv. Tutorials **21**(4), 3133–3174 (2019). https://doi.org/10.1109/COMST.2019.2916583
14. Marsch, P., Da Silva, I., Bulakci, O., Tesanovic, M., El Ayoubi, S.E., Rosowski, T., Kaloxylos, A., Boldi, M.: 5G radio access network architecture: design guidelines and key considerations. IEEE Commun. Mag. **54**(11), 24–32 (2016). https://doi.org/10.1109/MCOM.2016.1600147CM
15. Michalski, R.S., Carbonell, J.G., Mitchell, T.M.: Machine Learning: An Artificial Intelligence Approach. Springer Science & Business Media, Berlin (2013)
16. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937. PMLR (2016)
17. Nasir, Y.S., Guo, D.: Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks. IEEE J. Sel. Areas Commun. **37**(10), 2239–2250 (2019). https://doi.org/10.1109/JSAC.2019.2933973
18. Nguyen, V.G., Brunstrom, A., Grinnemo, K.J., Taheri, J.: SDN/NFV-based mobile packet core network architectures: a survey. IEEE Commun. Surv. Tutorials **19**(3), 1567–1602 (2017). https://doi.org/10.1109/COMST.2017.2690823
19. Qu, L., Assi, C., Shaban, K.: Delay-aware scheduling and resource optimization with network function virtualization. IEEE Trans. Commun. **64**(9), 3746–3758 (2016). https://doi.org/10.1109/TCOMM.2016.2580150
20. Raza, M.R., Natalino, C., Öhlen, P., Wosinska, L., Monti, P.: Reinforcement learning for slicing in a 5G flexible ran. J. Lightwave Technol. **37**(20), 5161–5169 (2019). https://doi.org/10.1109/JLT.2019.2924345
21. Rostami, A., Öhlén, P., Santos, M.A.S., Vidal, A.: Multi-domain orchestration across ran and transport for 5G. In: Proceedings of the 2016 ACM SIGCOMM Conference, pp. 613–614 (2016)
22. Saleem, Y., Yau, K.L.A., Mohamad, H., Ramli, N., Rehmani, M.H., Ni, Q.: Clustering and reinforcement-learning-based routing for cognitive radio networks. IEEE Wireless Commun. **24**(4), 146–151 (2017). https://doi.org/10.1109/MWC.2017.1600117

23. Sallent, O., Perez-Romero, J., Ferrus, R., Agusti, R.: On radio access network slicing from a radio resource management perspective. IEEE Wireless Commun. **24**(5), 166–174 (2017). https://doi.org/10.1109/MWC.2017.1600220WC
24. Samdanis, K., Costa-Perez, X., Sciancalepore, V.: From network sharing to multi-tenancy: the 5G network slice broker. IEEE Commun. Mag. **54**(7), 32–39 (2016)
25. Sciancalepore, V., Samdanis, K., Costa-Perez, X., Bega, D., Gramaglia, M., Banchs, A.: Mobile traffic forecasting for maximizing 5G network slicing resource utilization. In: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, pp. 1–9 (2017). https://doi.org/10.1109/INFOCOM.2017.8057230
26. Sciancalepore, V., Costa-Perez, X., Banchs, A.: RL-NSB: reinforcement learning-based 5G network slice broker. IEEE/ACM Trans. Networking **27**(4), 1543–1557 (2019). https://doi.org/10.1109/TNET.2019.2924471
27. Shen, X., Gao, J., Wu, W., Lyu, K., Li, M., Zhuang, W., Li, X., Rao, J.: AI-assisted network-slicing based next-generation wireless networks. IEEE Open J. Veh. Technol. **1**, 45–66 (2020)
28. Sulaiman, M., Moayyedi, A., Ahmadi, M., Salahuddin, M.A., Boutaba, R., Saleh, A.: Coordinated slicing and admission control using multi-agent deep reinforcement learning. IEEE Trans. Netw. Serv. Manage. **20**(2), 1110–1124 (2023). https://doi.org/10.1109/TNSM.2022.3222589
29. Sun, Y., Peng, M., Zhou, Y., Huang, Y., Mao, S.: Application of machine learning in wireless networks: Key techniques and open issues. IEEE Commun. Surv. Tutorials **21**(4), 3072–3108 (2019)
30. Taleb, T., Ksentini, A., Frangoudis, P.A.: Follow-me cloud: when cloud services follow mobile users. IEEE Trans. Cloud Comput. **7**(2), 369–382 (2019)
31. Tang, H., Zhou, D., Chen, D.: Dynamic network function instance scaling based on traffic forecasting and VNF placement in operator data centers. IEEE Trans. Parallel Distrib. Syst. **30**(3), 530–543 (2019). https://doi.org/10.1109/TPDS.2018.2867587
32. Tu, H., Zhao, L., Zhang, Y., Zheng, G., Feng, C., Song, S., Liang, K.: Deep reinforcement learning for optimization of ran slicing relying on control-and user-plane separation. IEEE Internet Things J., 1–1 (2023). https://doi.org/10.1109/JIOT.2023.3320434
33. Van Huynh, N., Thai Hoang, D., Nguyen, D.N., Dutkiewicz, E.: Optimal and fast real-time resource slicing with deep dueling neural networks. IEEE J. Sel. Areas Commun. **37**(6), 1455–1470 (2019)
34. Vincenzi, M., Antonopoulos, A., Kartsakli, E., Vardakas, J., Alonso, L., Verikoukis, C.: Multi-tenant slicing for spectrum management on the road to 5G. IEEE Wireless Commun. **24**(5), 118–125 (2017). https://doi.org/10.1109/MWC.2017.1700138
35. Ye, Q., Li, J., Qu, K., Zhuang, W., Shen, X.S., Li, X.: End-to-end quality of service in 5G networks: examining the effectiveness of a network slicing framework. IEEE Veh. Technol. Mag. **13**(2), 65–74 (2018). https://doi.org/10.1109/MVT.2018.2809473
36. Zhang, K., Yang, Z., Başar, T.: Multi-agent reinforcement learning: a selective overview of theories and algorithms. In: Handbook of Reinforcement Learning and Control, pp. 321–384 (2021)
37. Zong, B., Fan, C., Wang, X., Duan, X., Wang, B., Wang, J.: 6G technologies: key drivers, core requirements, system architectures, and enabling technologies. IEEE Veh. Technol. Mag. **14**(3), 18–27 (2019)

Chapter 5
Customized Slicing for Industrial Applications

5.1 5G-Enabled New Industrial Scenarios

New requirements of diverse use cases, in particular, for vertical industries, such as vehicle to everything, smart utilities, and Industry 4.0, are accelerating the maturity and commercialization of 5G wireless communication systems [11]. Historically, the Industrial Revolution critically hinges on the rapid development of communication technology (CT). 5G techniques support massive connectivity, high-rate data transmission, low latency, and high reliability, enabling some critical use cases of Industry 4.0 [27]. As more and more vertical industries join the business model of wireless systems, emerging applications in the industrial space are now creating new market opportunities to MNOs. Meanwhile, enabling real-time remote control and high-safety information exchange bring more challenges to wireless networking in industrial scenarios.

Communication technology needs to form a joint force with industrial automation technology to realize the intelligent operation of industrial enterprises in all aspects of design, procurement, production, warehousing, logistics, operation, and sales. As the foundation of the industrial Internet, the network needs to have the ability to access massive devices, interworking of heterogeneous systems, end-to-end deterministic transmission, and intelligent scheduling of network resources. 5G is the key enabling technology of the industrial Internet, and the industrial Internet is one of the important application scenarios of 5G. Here we summarized some typical use cases in industrial scenarios and analyzed the difference in requirements of these use cases, providing basis for customized resource scheduling in 5G-enabled new industrial scenarios.

W. Guan, H. Zhang, *Network Slicing for Future Wireless Communication*, Wireless Networks, https://doi.org/10.1007/978-3-031-58229-5_5

5.1.1 Use Case Requirements and Smart Industry

The industrial applications mainly related to the operation in three domains, data collection, storage and analysis in the information technology (IT) domains, the productive activities in the operation technology (OT) domains, and interconnection among a massive number of the existing industrial elements in the CT domain [27]. The communication networks (CT domain) are the key of realizing the interaction between the physical world (OT domain) and the digital world (IT domain), and CT-IT-OT (CIOT) collaboration is the foundation of Industry 4.0.

According to the 5G Alliance for Connected Industries and Automation (5G-ACIA) white paper [29], some new and evolved industrial use cases for Industry 4.0. are described below.

- **AI-based quality inspection.** This use case for optimizing product quality involves collecting vision information through high-definition industrial cameras, using the AI-based algorithms to analyze the images or video received and detecting the quality of the products on production in near-real-time. The requirements of AI-based quality inspection include higher sensitivity, higher precision, and higher efficiency, which aims at achieving better performance than the traditional manual quality inspection and reducing the human cost.
- **XR-aided industry manufacturing.** This use case is built on the collaboration between humans and machines, using virtual reality, augmented reality, and mixed reality (XR) technologies to achieve flexible production. With the remote assistance and guidance, machines can perform repetitive tasks precisely and the safety of human workers can be ensured. XR-aided industry manufacturing requires ultra-low latency and high reliable communications between the sensors installed on the machines and the XR devices.
- **Remote control and manipulation.** This is an existing use case in 3GPP TR 22.804 [1], where a remote control center or diagnostic system performs the required manipulation based on the periodic access to the huge volume of sensor data. In the evolved scenario, the main prerequisites for this use case are a device-to-network endpoint latency of about 5 ms, high service reliability with an uplink speed of 3 to 8 Mbit/s for 1080 pixel images, and a remote control bit rate of 100 kbps.
- **Predictive monitoring and maintenance.** This use case is expected to avoid the sudden failure of the production line by leveraging IoT sensor networks, machine learning algorithms, and big data analysis. Condition monitoring and predictive maintenance in the context of industrial production require greater efficiency for aggregating locally sensor data, reliable communications for video surveillance service, and quick offloading of heavy computations.

The tight requirements of these use cases bring many challenges in realizing Industry 4.0, especially the challenges in the CT domain. First, due to the increase number of sensors and industrial devices, massive access to RAN brings challenge in improving the coverage and reducing energy consumption. Second, ensuring the

high quality of information transmission in the 5G-enabled industrial Internet brings challenge in reducing the E2E latency and providing high reliability. Last but not least, diverse applications in the industrial scenario which have different properties and requirements of resources, which brings challenge in efficient slice management. To deal with these challenges, many efforts have been devoted by academia and industry, such as the edge computing for delay-sensitive applications [15], reconfigurable intelligent surfaces for improving the channel conditions [7] and intelligent AI-based network slicing for reducing management costs [25]. However, most of the current researches in the area of 5G networks focus on the consumer Internet, which are difficult to meet the strict requirements of services in industrial Internet for network security, reliability, and certainty.

Deploying 5G networks in the industrial scenario means that the traffic flows of diverse industrial applications will be delivered in the wireless domain. The management of wireless resources should match the differentiated characteristics of industrial traffic flows. The data flows of multiple industrial applications transmitted in the wireless domain can be classified into three different traffic classes according to 3GPP TR 22.804: sporadic burst traffic, periodic time-sensitive traffic, and non-deterministic traffic. Some selected representative applications are summarized in Table 5.1 along with their requirements.

- **Sporadic burst traffic (SBT)** relates to applications such as emergency stops or failure alarms which are triggered when specific events or errors occur.
- **Periodic time-sensitive traffic (PTT)** which is generated periodically with a given deadline relates to most common industrial applications such as motion control and cooperative control.
- **Non-deterministic traffic (NDT)** is characteristic of applications such as software updates or user interaction, which do not require delay guarantee. Moreover, eMBB traffic supported by the Base Station (BS) in industrial environment is also classified as non-deterministic traffic.

Table 5.1 Typical traffic flows of industrial applications with requirements

Traffic class	Application	Latency	Reliability	Data rate	Payload
Sporadic burst traffic	Emergency stops	<4 ms	–	–	40–250 bytes
	Failure alarms	<50 ms	–	–	10–100 bytes
Periodic time-sensitive traffic	Motion control	2 ms	99.99–99.9999 (%)	–	20 bytes
	Cooperative control	1 ms	99.999–99.99999 (%)	–	40–250 bytes
Non-deterministic traffic	Software updates	–	–	>1 Mbps	–
	User interaction	–	–	>5 Mbps	–

The SBT can be generated at any point in time and the packets with a given payload should be successfully delivered before the latency deadline. The PTT is characterized by a transmission period in addition to the given payload and the latency deadline. The NDT only requires a number of RBs to achieve the data rate demand.

5.1.2 Standards and Techniques of IEEE TSN and 5G ULL

Although the Third Generation Partnership Project (3GPP) Releases 15 and 16 standards have introduced the matured 5G technologies for enhanced Mobile Broadband (eMBB) communications, wireless networks are still required to incorporate with deterministic communications to support ultra-Reliable Low-Latency Communications (uRLLC) services, especially the time-critical applications in industrial environments.

The standardization within IEEE 802.1 Time-Sensitive Networking (TSN) task group (TG) enables Ethernet to be a reliable real-time communication network which can simultaneously satisfy the demands of multiple time-critical applications as well as nontime-critical applications. The TSN standards define mechanisms about synchronization, bounded low latency, reliability, and resource management to provide deterministic services in many industries. Nasrallah et al. [19] provides a comprehensive up-to-date survey of the IEEE TSN standards, the Internet Engineering Task Force (IETF) Deterministic Networking (DetNet) standards, and the ULL research studies.

As a collective name for a set of standards, TSN standards were started as IEEE 802.1 Audio Video Bridging (AVB) and are successfully deployed for various industrial applications nowadays. The set of TSN standards can be divided into different clusters according to the functionalities of them, i.e., synchronization, latency, reliability, and resource management. Since some standards contribute to more than one aspect, these clusters are not disjoint [17]. In this subsection, some core standards for critical communication are briefly revived, i.e., the clock synchronization capabilities of TSN (IEEE 802.1AS/ASrev), frame preemption of low-critical frames by high-critical frames (IEEE 802.1Qbu-2016), time-triggered (TT) communication (IEEE 802.1Qbv-2015), as well as traffic filtering and policing (IEEE P802.1Qci-2017).

Time synchronization plays a crucial role for time-sensitive applications, thus a suite of clock synchronization protocols in IEEE 802.1AS is defined to ensure that end stations and bridges may synchronize their local clocks to each other. The main problems to be solved are determination of a synchronization hierarchy in the network, distribution of the time from the one or multiple grandmasters in the hierarchy to the rest of the network, and the measurement of link delays between devices.

Frame preemption is introduced in IEEE 802.1Qbu to address the problem that the transmission of urgent frames is prevented by the ongoing transmission

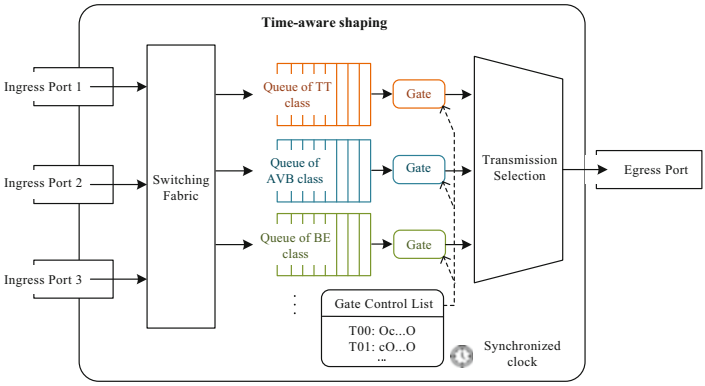


Fig. 5.1 The TAS in a TSN switch

of noncritical frames. Preemption reduces the transmission delay of time-critical frames because the critical data can be transmitted directly without waiting for the transmission of the noncritical data. Note that preemption occurs only if the preempted traffic support preemption and the impact of preemption is acceptable.

Except for the preemption mechanisms, the TT paradigm based on the clock synchronization protocols is also indispensable to the realization of deterministic communications. The core principle of TT communication is finding a feasible communication schedule which instructs the TSN end stations when to send which frames [26]. To realize this schedule, TSN adopts gate control list (GCL) in finding the right points in time when to enable and disable the transmission of TT and other traffic classes.

Realizing TT transmissions is mainly thanks to IEEE 802.1Qbv Time-Aware Shaper (TAS) and the standards IEEE 802.1ASrev which defines a time synchronization protocol. Figure 5.1 shows the TAS in a TSN switch. Instead of scheduling frame transmissions directly, TAS as a gate mechanism schedules the activation and deactivation of the traffic class queues which have different priorities. Traffic shaping comprises how frames are assigned to queues, and how and when frames are selected for transmission [17]. The point in time when to set the gate state of each queue into the open/closed state is decided by the GCL. A suite of clock synchronization protocols are defined such that end stations and bridges may synchronize their local clocks to each other. Losing synchronized time may result in transmission error.

In TSN switch, the switching fabric identifies the frames based on the information in the frame header. The buffered frames are distributed on multiple first-in-first-out queues based on the priority code point bits in the headers. Different traffic classes are isolated in separate queues and each queue is associated with a gate controlled by GCL. GCL indicates the open/close state for a certain time window and state changes are statically scheduled with respect to a synchronized time. The traffic classes are prioritized by TAS and the deterministic nature of TT traffic is

guaranteed by finding a GCL, i.e., the points in time when to change the gate state for each queue.

During queuing frames and transmission selection, traffic policing is done by ingress filtering, egress filtering, as well as flow metering. The IEEE 802.1Qci standard defines protocols and procedures to make filtering, policing, and queuing decisions. Queues with higher priority are served before queues with lower priority. It provides for quality of service protection and avoids the interference when multiple streams share the same switch egress queue by using stream identifier and priority.

As a key technology of providing deterministic guarantees for Ethernet-based communications, TSN technology standardized in IEEE 802.1 is integrated in 5G System (5GS) to enable the simultaneous transmission of deterministic traffic and eMBB traffic [24]. In the 5G wireless context, the support of deterministic traffic has also been discussed, especially for traffic with deterministic end-to-end ULL requirements [19]. The DetNet working group (WG) focuses on layer 3 routed segments while the TSN TG focuses on layer 2 bridged networks.

DetNet flows are specified by their QoS classes which are defined by he maximum and minimum end-to-end latency, and the packet loss probability requirements [9]. To guarantee the QoS of DetNet flows and ensure that the non-DetNet flows have no effect on DetNet flows, the DetNet flows are mainly divided into four types and each DetNet flow is identified based on the flow ID and DetNet Control Word. The time synchronization between DetNet capable network entities is ensured through various existing synchronization techniques, e.g., IEEE 802.1AS. To support minimal jitter, i.e., extremely low delay variations, DetNet specifies jitter reduction through sub-microsecond time synchronization and time-of-execution fields embedded within the application packets [9].

The DetNet WG mainly focused on flow management which specifies DetNet configuration model and resources distribution for DetNet flows, and flow integrity which protect DetNet flows against possible failures through packet replication and elimination function (PREF) and fault mitigation. Following the same principles used for IEEE TSN TG deterministic flows, DetNet flow control identifies the data and control plane solutions, and defines queuing, shaping, scheduling, and preemption principles to achieve deterministic bounded latency and packet loss. Current research studies in aspect of DetNet focus on the flow control and flow integrity. For industrial applications that require deterministic characteristics, the routing mechanisms [20] and the scheduling mechanisms [13] are proposed to enable determinacy. In addition to the IEEE and IETF standardization organizations, 3GPP and ETSI also contribute to the development of 5G ULL standardization. Reducing the latency in the wireless access segment [18] and addressing ULL in the fronthaul [5], backhaul [14] attract many concerns in current 5G ULL research studies.

5.2 QoS-Aware Traffic Scheduling Toward New 5G Capabilities

Since most of industrial services need to be carried by a network with deterministic transmission and reliability guarantee, Time-Sensitive Network (TSN) has attracted extensive attention due to its delay determination, and forwarding capability and compatibility with Ethernet protocols. However, wired TSN cannot meet the extensive deployment needs of new devices such as massive sensors and AGVs in smart factories. Thus, the collaborative transmission of 5G and TSN has become an important foundation for realizing wireless industrial Internet and flexible manufacturing. Although the interaction between 5GS and TSN is already part of the 3GPP specifications, few researches focus on the QoS-aware traffic scheduling in 5G-TSN integration. Here we provide a brief review of three typical scenarios of 5G TSN integration and the QoS-aware traffic scheduling in 5G network slicing, aim at find an efficient solution of scheduling 5G traffic flows and time-critical traffic flows simultaneously in the scenarios of 5G TSN integration.

5.2.1 Technical Directions of 5G TSN Integration

According to 3GPP specifications, a set of new functionalities have been incorporated in 5GS architecture to connect TSN control plane with 5GS. There are three typical scenarios of integration between 5GS and TSN shown in Fig. 5.2, first is TSN over 5G uRLLC, second is 5GS as a TSN bridge entity, and the last one is using TSN in 5G Xhaul network (e.g., fronthaul, midhaul, and backhaul).

- **TSN over 5G uRLLC.** In this scenario, TSN and 5G network are deployed jointly. In other words, the original time-sensitive service system (such as industrial control network, vehicle network, etc.) is connected to the 5G system directly, and 5G uRLLC is used to increase the coverage distance of the TSN system. Multiple types of traffic flow are scheduled synergistically, and the quality of E2E service delivery are guaranteed by realizing the deterministic of transmission in segments.
  As shown in Fig. 5.2a, the entire service system can be regarded as a UE. It is needed to establish a mapping relationship between the traffic classification in TSN and the service type of the 5G network. At the same time, it is necessary to retain the relevant marks of the traffic configuration of TSN and strip the 5G package after the remote transmission of the 5G network. After entering the cooperative service system, the deterministic transmission is still carried out following the way of scheduling the TSN traffic.
- **5GS as a TSN bridge entity.** As described in 3GPP R16 23.501, the entire 5G network has been upgraded to carry the deterministic transmission of TSN traffic. In this scenario shown in Fig. 5.2b, 5GS can be regarded as a TSN bridge entity

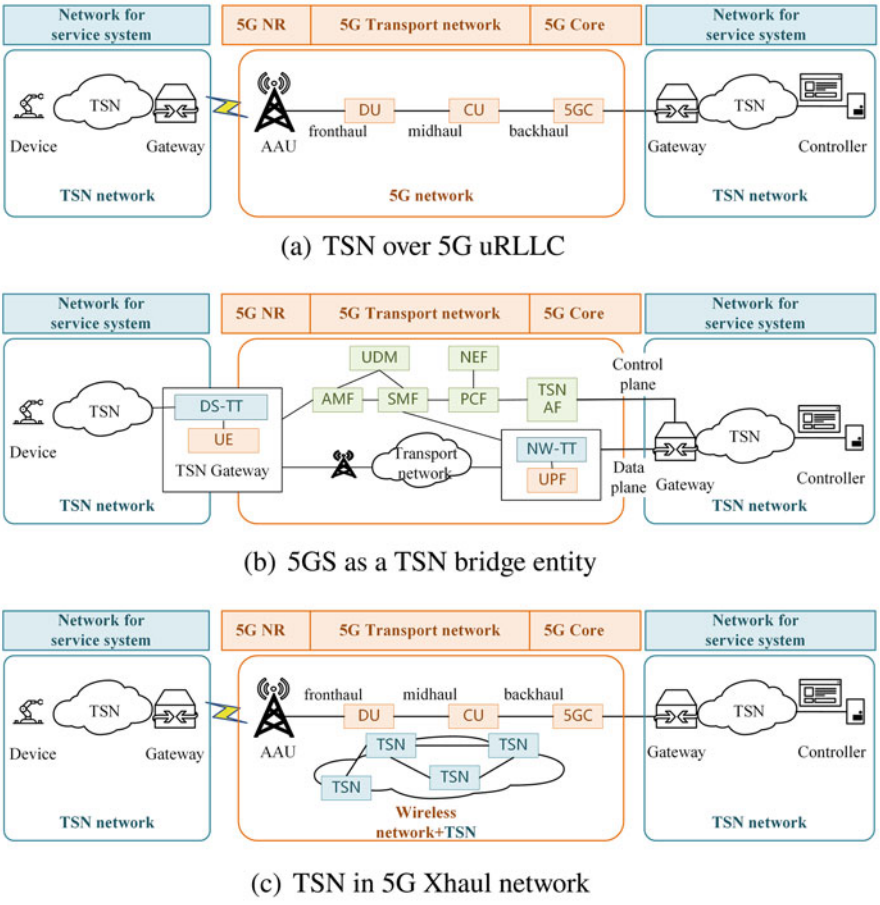


Fig. 5.2 Three typical scenarios of integration between 5GS and TSN

which support TSN centralized architecture and time synchronization mechanism. Moreover, accurate traffic scheduling is achieved by defining new QoS models (flow direction, cycle, burst arrival time), thus high quality simultaneous transmission of multi-type deterministic traffic between UE and UPF in 5GS is realized.

Realizing this scenario depends on the support of three aspects of techniques. First is the integration of TSN and the air interface in 5G which is also named 5G New Radio (NR). The definition of time synchronization, latency and delay jitter should be added in uRLLC communication. Secondly, device-side TSN translator (DS-TT) and network-side TSN translator (NW-TT) should be deployed, supporting mappings between UE and service systems for ports, protocol data units and QoS mechanisms, and TSN-related traffic scheduling

features. Last is the ability to enable deterministic communication between UE and UE under the same UPF.

- **TSN in 5G Xhaul network.** In addition to 5G NR standards and new core network architecture, the reconstruction of 5G transport network (TN) which interconnects RAN and 5G core network is also an important research direction. The network functions (NFs) of RAN reside in the Central Unit (CU), Distributed Unit (DU). The different TN segments interconnecting these NFs and the other NFs of 5G core, which can be denoted as fronthaul, midhaul, and backhaul, respectively.
  Using TSN network to improve the quality of 5G TN is attracting the attentions of researchers [4]. TSN can be used to support internal 5G TN operations for the fronthaul, but also for the entire converged Xhaul network. Driving the large-scale deployments of TSN-aware 5G TN relies on the outcome of the liaison activities between IEEE TSN and IETF DetNet. Notably, when the TSN network serving the Xhaul, supporting 5G traffic flows and other unpredefined traffic flows will increase the complexity of TSN network optimization.

In the industrial scenario, the difficulty and key point of 5G TSN integration is how to achieve deterministic transmission on 5G network and ensure the SLA requirements of different traffic flows in the industrial Internet. Promoting the rapid and low-cost deployment of 5G BSs in the industrial Internet scenario requires to extend TSN capabilities over a wireless network. The 3GPP standardization body is continuously working on the evolution of 5G technologies, especially about the uRLLC. In order to support low-latency communication, 5G NR defines a flexible frame structure and mini-slot transmissions based on different numerologies [16], supporting mini-slots which comprise 2, 4, or 7 OFDM symbols. In mini-slot time scale, transmission can start immediately without needing to wait for slot boundaries, which enables quick delivery of low-latency payloads.

To efficiently support both broadband and ultra-reliable low-latency communications, the 3GPP standards body has proposed an innovative superposition/puncturing framework for multiplexing URLLC and eMBB traffic in 5G [2]. The NR numerology, mini-slot and the superposition/puncturing framework for joint eMBB and uRLLC traffic is shown in Fig. 5.3. In the puncturing framework, time is divided into slots and further subdivided into mini-slots. eMBB traffic is scheduled at the beginning of slots and share the bandwidth over the time-frequency plane. Once the uRLLC packet arrives, they can be immediately scheduled in the next mini-slot on top of the ongoing eMBB transmissions. Superposition refers that the BS chooses non-zero transmission powers for both eMBB and overlapping URLLC traffic. Puncturing means that only eMBB transmissions are allocated zero power when URLLC traffic is overlapped.

Besides the mini-slot transmission, technologies of network slicing and edge computing in 5G are also interest future industrial applications. The 3GPP Release-16 extends support to apply TSN in 5G, focusing on the major vertical area Industrial Internet of Things (IIoT). As a promising technique to accommodate diverse services for the IIoT, network slicing has been studied in a large number

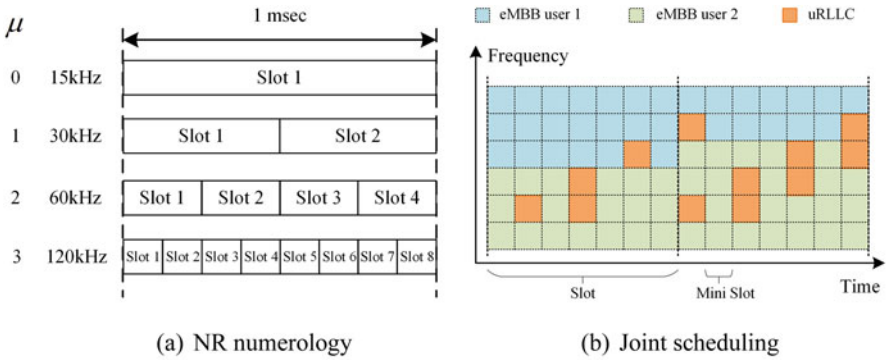


Fig. 5.3 NR numerology, mini-slot and the superposition/puncturing framework for joint eMBB and uRLLC traffic

of current literature. Authors in [30] present an architecture of intelligent network slicing management for IIoT applications and summarize the existing works on network slicing for three IIoT applications, smart transportation, smart energy, and smart factory. By offloading data from IIoT cloud data centers to edge networks, edge computing applied in IIoT enables to improve performance of data processing, protect data security and privacy of enterprises, reduce the total task execution cost (including energy and delay). In [22], authors outline the research progress concerning edge computing in IIoT and introduce a reference architecture of edge computing in IIoT.

5.2.2 *QoS-Aware Traffic Scheduling in Network Slicing*

Integrating TSN functionalities in the 5GS is essential for the widespread deployment of 5G in the IIoT, and the "black box" approach that 5GS integrates with TSN as a logical bridge is a promising method to accelerate the deployment. In this approach, the core network and RAN procedures remain hidden from the TSN network, and TSN standards can be developed independently without having a strong coupling with 3GPP standardization efforts and timetables. Although the TSN translator functionality for both the user plane and the control plane enables the interoperation between TSN and 5GS, supporting the simultaneous transmission of TSN traffic and 5G traffic requires a QoS-aware traffic scheduling method for 5G RAN. It is challenging that optimizing radio resources utilization while satisfying the QoS demand of TSN traffic and 5G traffic.

Network slicing enables the services with partially conflicting objectives to be accommodated efficiently within the same infrastructure network. A slice which contains multiple network functions and virtual resources can be created dynamically [6], scaled up or down with more or fewer resources [31], and

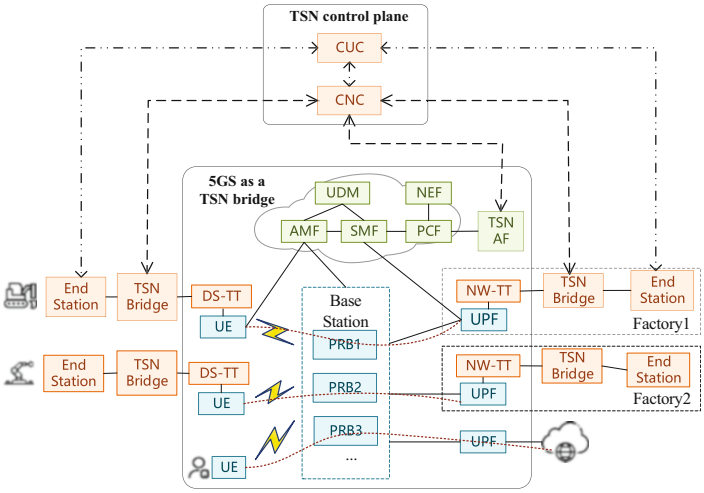


Fig. 5.4 The architecture of RAN slicing in 5GS exposed as a TSN logical bridge

reconfigured by adding or removing network functions [28]. At present, the dynamic management and orchestration of slices has been thoroughly investigated. Because of the uncertainty of radio channel, resource scheduling in RAN slicing faces greater challenges than core network slicing. Especially in the 5GS as a TSN logical bridge, partitioning radio resources for different type of slices needs to take into account the differentiated requirements of diverse TSN services and 5G services.

Figure 5.4 shows the architecture of RAN slicing in 5GS exposed as a TSN logical bridge, where a fully centralized TSN network is considered. As described in IEEE 802.1Qcc, there are two key elements in the configuration models of TSN control plane, the Centralized Network Configuration (CNC) entity and the Centralized User Configuration (CUC) entity. CNC which has a complete knowledge of the network topology and all the data flows is responsible for configuring TSN features and performing operations required for frame preemption and TAS at the TSN bridge. CUC is responsible for discovering end stations, retrieving the capabilities of end stations and configuring TSN features in end stations.

As shown in Fig. 5.4, TSN translation functionalities are embedded for both user and control planes inside the 5GS. In the user plane, NW-TT is deployed in the User Plane Function (UPF) and DS-TT is deployed in the User Equipment (UE). Both of them are used to perform QoS mappings, execute Per Stream Filtering and Policing (PSFP) functionality, and support hold and forward functionality. Application Function (AF) in the control plane interacts with the CNC entity and obtained PSFP information to achieve transmission reliability. The CNC entity sends the stream specification received from the CUC entity and the configuration information of 5G-TSN bridge to AF. Conversely, AF sends the feedback and reports flow QoS to the CNC. Moreover, NW-TT and DS-TT support the TSN time-

domain grandmaster clock while the rest of the 5GS components like UE, gNB, etc., are synchronized with the 5GS grandmaster clock.

In the BS of 5GS as TSN logical bridge, slice scheduler allocates physical resource blocks (PRBs) to different types of traffics, including nontime-critical traffic for eMBB users and time-critical traffic for uRLLC users in different time scales. Since slices need to conform to specific SLAs of different services, authors in [23] presented a utility-based inter-slice scheduling algorithm for three types of slices with specific QoS requirements. In order to satisfy the latency requirements of uRLLC slices while sharing radio resources with other types of slices, a two-level medium access control (MAC) scheduling solution is utilized in [3]. First level is slice-specific scheduling, with which the virtualized resources, i.e., frequency bandwidth, are assigned to UEs. Second-level is assigning physical resources to UEs based on the results of the first level and performing inter-slice resource partition. To satisfy the heterogeneous requirements of the eMBB and uRLLC services, authors in [8] propose a DRL-based approach to efficiently allocate radio resources in two different time scales. Despite the effectiveness of these RAN slice scheduling approaches, slicing the radio resources in 5G-TSN integration is expected to be further explored.

5.3 Customized RAN Slicing for 5G-TSN Integration

As a crucial innovation in 5G, network slicing offers the ability of tailoring resources on demand, which guarantees traffic isolation and improves resource utilization. By customizing the resource allocation for different services, RAN slicing enables fine-scale resource sharing among different service providers [12]. To transmit deterministic traffic and eMBB traffic in the same RAN, a suitable RAN slicing method for the scenario of 5GS as a TSN bridge is required. The RAN slicing method proposed in [10] meets the delay demand of deterministic traffic through resource reservation and preemption. The preemption-based methods allow the time-critical traffic to interrupt ongoing nontime-critical transmissions, which is appropriate to the transmission of urgent frames [17]. In this section, an AI-enabled resource slicing method is introduced, guaranteeing the latency requirements of time-critical traffic based on the reasonable preemption.

5.3.1 Deterministic Transmission in 5G-TSN

Deterministic latency and high-reliability performance are the prerequisites of satisfying the requirements of various industrial applications. In a TSN network, nodes (e.g., switches and end stations) communicate with each other by transmitting a stream of frames from the sender to the listener. For each stream with specific information including the bounded latency and jitter, the data size and the period, it

is assumed that the sender and receiver nodes as well as the routed communication path are known and given. In the classical approach, the sender node of a TT frame is configured with a scheduled transmission point in time which relates to a reference time established by a synchronization protocol. The TT switching is discussed previously, providing more capabilities for scheduling TT traffic.

For the industrial scenario of 5G-TSN integration, diverse QoS requirements of time-critical and nontime-critical traffic flows on a wireless TSN BS need to be guaranteed simultaneously. Similar to multiplexing uRLLC and eMBB traffic in 5G, supporting the coexistence of time-critical TSN traffic and nontime-critical 5G traffic in the 5G-TSN integration also need resource preemption. By puncturing any mini-slots of slots which has been allocated to nontime-critical 5G traffic, time-critical TSN traffic can be transmitted immediately. The wireless TSN BS allocates zero power to nontime-critical 5G traffic transmissions when time-critical TSN traffic is puncturing. The locations of TSN traffic puncturing can be used for nontime-critical users to decode transmissions. There will be some possible loss of rate caused by puncturing.

Since not all nontime-critical users can tolerate the impair of resource preemption, the nontime-critical traffic is classified into two types, non-preemptable traffic and preemptable traffic. Although resource preemption ensures the timely transmission of time-critical traffic, constantly high preemption is inefficient for the scenario where many time-critical applications are running in the industrial terminals. Hence, instead of using a solely preemption-based approach, restricting preemption and reserving resource for deterministic traffic achieve more efficient resource utilization [10]. Since over-provisioned reservation might result in more unused resources, finding an optimal resource allocation for each slot is of great concern.

To realize the next generation industrial wireless communication, customized resource slicing is absolutely necessary to guarantee diverse QoS requirements of time-critical TSN traffic and nontime-critical 5G traffic flows on a wireless TSN BS. Partitioning the radio resources among RAN slices needs to consider not only the requirements in terms of data rate, reliability, and bounded latency but also the priority level and arrival time of traffic flows. Taking industrial wireless network as an example, the AI-based time-sensitive RAN slicing framework for 5GS as a TSN logical bridge is elaborated in Fig. 5.5. The integral architecture consists of three parts: physical infrastructure bearing the traffic transmission, scheduling process of multi-type traffic, service and application corresponding to the specific traffic.

In the envisioned framework, the wired TSN segment includes a TSN switch and TSN end stations (e.g., controller and factory devices). DS-TT and NW-TT provide support for TSN ingress and egress ports, enabling the interoperation between TSN and 5GS. The wireless domain exhibits the user plane which carries user traffic exchanged between UE and gNB. A protocol data unit (PDU) session between the UE and the TSN switch is established to enable time synchronization and support connectivity to the TSN domain.

In this framework, fog nodes with differing computation and storage capabilities are connected with each other and to the cloud. Based on the work in [21],

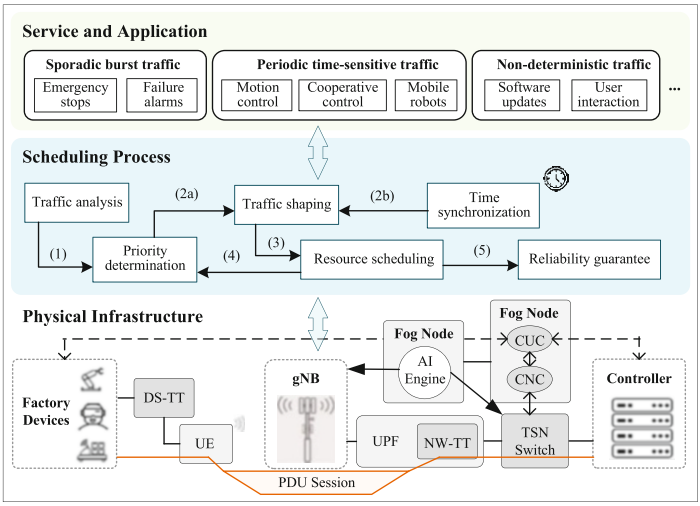


Fig. 5.5 The AI-based time-sensitive RAN slicing framework

some of the fog nodes can be used for the implementation of CUC and CNC, which performs complex computational tasks of deriving the schedules for TSN switch. Additionally, the fog nodes can be equipped with an AI-engine occupying partial computing and storage resources. AI-Engine which also can be deployed as an independent physical entity encapsulates diverse machine learning models for solving complex problems. An AI-engine which encapsulates diverse ML models is designed in the proposed framework to provide intelligent solutions for many use cases. For example, AI-engine provides DRL models to make optimal decisions for wireless resource allocation, supervised learning models to predict the changes of data traffic, object detection models to support intelligent applications in the edge [32].

The AI-engine is designed by us in [32] and the structure of AI-engine is shown in Fig. 5.6. In order to realize the flexible utilization of the AI-engine, a distributed deployment approach is adopted based on GPU virtualization. There are some distributed components of AI-engine deployed in the edge or the cloud to meet different requirements. Various AI algorithms are encapsulated in AI-engine to provide intelligent support for many scenarios, such as DRL algorithms that support life-cycle management of RAN slices, recurrent neural network (RNN) that enhances the capability of network data analytics function (NWDAF) in the core network, and Yolo that supports object detection in the edge cloud. The wireless communication system gains support from AI-engine through a common interface. The related parameters required for ML models are sent to AI-engine and the optimal decisions or analysis results are received.

In 5G-TSN integration, how to realize time-aware scheduling for the time-critical traffic cross the wired and wireless domains, as well as allocate resource for TSN and 5G services with heterogeneous requirements is challenging. In the case of

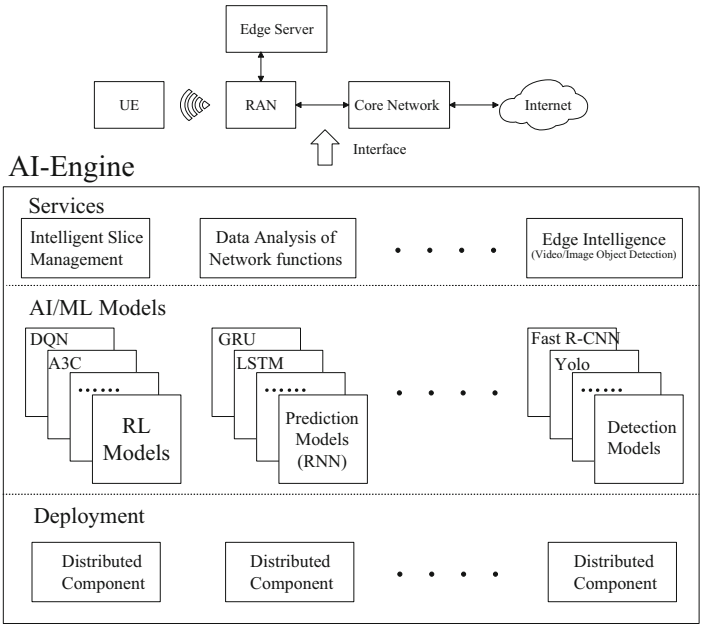


Fig. 5.6 The structure of AI-engine

transmitting streams from controller to factory devices, the steps of scheduling process are illustrated in Fig. 5.5.

(1) Once CUC received stream requests from end stations, the traffic characteristic parameters such as source-destination pair, period, payload, and deadline are obtained and communicated to CNC. AI-Engine is utilized to provide AI algorithms for CNC to perform traffic analysis, including the traffic classification, flow prediction, and QoS extraction. After the traffic analysis, each traffic is uniquely identified with a priority level.

(2) In the CNC, the communication schedule between the different streams is computed. The configuration messages of all streams are originated and distributed to the TSN switch. According to the priority level and delay budget of flows, the frame transmission over the egress port in a TSN switch is controlled by traffic shaping as depicted before. To obtain appropriate time points for all streams, TSN switch is synchronized to network timing.

(3) The schedule is useful for gNB and allows it to efficiently allocate resources to upcoming traffic flows in a real-time manner. DRL models encapsulated in AI-Engine are used for gNB to realize RAN slicing. RBs are assigned to different slices, carrying out the traffic transmission for both TSN services and 5G services. AI-assisted resource scheduling approach is introduced in Sec. V to satisfy the heterogeneous requirements of SBT, PTT, NDT.

(4) In order to achieve the performance target of time-sensitive traffic, results of wireless resource scheduling are feedback to dynamically change the priority level of upcoming traffic flows. Dynamic priority setting allows to change the priority level based on the waiting time or the transmission delay of a stream compared to the predefined threshold. Adjusting the schedule based on the network status is allowed, which prevents prolonged delays for low priority traffic and keeps the worst-case delay within prescribed limits.
(5) To avoid the effect of wireless channel condition, hybrid automatic repeat request (HARQ) techniques and redundant traffic transmission are enabled to guarantee high reliability for periodic traffic. As a means of active redundancy, HARQ shortens duplicate frames and minimizes radio resource consumption, but the achievable latency is affected by waiting for the ACK from the receiver. On the contrary, passive redundancy is realized by transmitting different copies of the same frame through at least two disjoint paths.

5.3.2 AI-Enabled Resource Slicing for 5G-TSN

The current works in RAN slicing explored many approaches of resource scheduling and eMBB/uRLLC multiplexing based on assumptions of specific mathematical models. Due to the inaccuracy and the ever-increasing complexity of model-based approaches, model free AI-assisted solutions are studied in some researches. In the proposed RAN slicing framework, AI-engine provides some DRL-based algorithms to solve the real-time resource scheduling problem.To illustrate the AI-engine assisted radio resource management in 5GS as a TSN logical bridge, a DQL-based resource scheduling is introduced in this section.

RAN slices are requested for both the traffic transmitted from the TSN translator and the local traffic in 5GS. As for the six applications mentioned in the Table 5.1, there will be three categories of slice requests, and each category includes two different slices serving users with the similar QoS requirements, respectively. These slice requests are different in latency requirements and data rate demand. Maximizing the resource utility while satisfying the distinct requirements of these slice requests is challenging.

S_n denotes a set of NDT slice requests. For each NDT slice request s_i, to satisfy the demand of M users, the number of RBs allocated to s_i can be denoted as

$$Z_i = \sum_{u=1}^{M} \left\lceil \frac{R_i}{R_u^e} \right\rceil, \forall s_i \in S_n, \tag{5.1}$$

where R_i is the target transmission rate of slice request s_i. R_u^e is the effective transmission rate or throughput that the user u will experience for each RB assigned, which is related with the Signal to Interference plus Noise Ratio (SINR) and the Block Error Rate (BLER). Each NDT slice is require to allocate Z_i RBs during the

transmission window T_w. Defining $O_{i,t}$ to represent the number of RBs allocated to NDT slice s_i in slot t, thus we have

$$\sum_{t=1}^{T_w} O_{i,t} = Z_i \tag{5.2}$$

S_p denotes a set of PTT slice requests. For each PTT slice request s_i, the transmission period is T_p^i and b_i bits of data need to be transmitted in D_i. Hence, the transmission rate is

$$R_i = \frac{b_i}{D_i}$$

and the number of RBs allocated to s_i can be denoted as

$$Z_i = \sum_{u=1}^{M} \left\lceil \frac{R_i}{R_u^e} \right\rceil, \forall s_i \in S_p. \tag{5.3}$$

Each PTT slice is required to reserve Z_i RBs within the transmission window T_p^i. To ensure the delay requirements, the time slots of these RBs need to meet the conditions

$$\sum_{t=t_z}^{t_z+D_i-1} O_{i,t} = Z_i, \forall t_z \in T_0, \tag{5.4}$$

where $T_0 = \left\{ t_z \middle| t_z = t_0 + zT_p^i, \forall z \in \left\{ 0, 1, \ldots, \left\lfloor T_w \middle/ T_p^i \right\rfloor - 1 \right\} \right\}$. t_0 represents the time of the first transmission of data and t_z represents the time when packet z is generated. This constraint guarantees that data can be transmitted within the deadline D_i.

S_s denotes a set of SBT slice requests. For each SBT slice request s_i, b_i bits of data need to be transmitted within the delay D_i with reliability P_r as the success rate. The data of SBT slice is generated in random, when the number of users is M, assuming that the packets generated by each user follow the Poisson distribution with exponential arrival time and λ the average number of packets generated per second, then the average arrival time of packets is $\frac{1}{\lambda}$, the probability of generating packets within the time interval T_{slot} is $P_p = 1 - e^{(-T_{slot}\lambda)}$, and the probability of packet loss is

$$P_m = 1 - \left(\frac{k - \overline{J} P_p}{k} \right)^{M-1}, \tag{5.5}$$

where $\overline{J} = \frac{1}{M} \sum_{u=1}^{M} \left\lceil \frac{R_i}{R_u^e} \right\rceil$. and reliability guarantee is $P_r \geq 1 - P_m$. The minimum number of RBs required to meet reliability requirements is k, and the number of RBs required to serve M users is

$$Z_i = \min \left(k, \sum_{u=1}^{M} \left\lceil \frac{R_i}{R_u^e} \right\rceil \right), \forall s_i \in S_n. \tag{5.6}$$

The data requested by the SBT slice needs to be satisfied within the delay D_i, so the condition for the time slots of Z_i RBs is

$$\sum_{t=l}^{l+D_i-1} O_{i,t} = Z_i, \forall l \in [1, T_w]. \tag{5.7}$$

Based on the demand characteristics of these requests, the proposed multi-slice scheduler comprises three major steps:

1. Due to the high-reliability requirements, resources are reserved for the slices that support the PTT firstly.
2. Then, following the data rate demand of slices that support the NDT, the remaining resources are allocated to the preemptable slices and non-preemptable slices.
3. Finally, the SBT preempts the resource allocated to the preemptable NDT slices on a mini-slot scale.

For the downlink scheduling, the information of the incoming slice requests are obtained at the beginning of each allocation window. The number of RBs required by slices that support the NDT is decided by the data rate demand and the experienced throughput per assigned RB. Also, whether this slice supports resource preemption is known. Given the transmission period of the slices that support the PTT, the initial slot to transmit data is ascertained. The number of RBs required by these slices is decided by the payload and the latency deadline based on the experienced throughput per assigned RB. As we have no knowledge of whether or not and when the SBT will arrive within this allocation window, the resource requirements of the SBT are satisfied by preemption on the slices for the preemptable NDT. In 5G NR, each time slot is divided into a number of mini-slots and this article considers seven mini-slots within a time slot. Once the SBT arrives, the next mini-slot is scheduled for its transmission without waiting for the end of NDT transmissions. The number of subcarriers and mini-slots required by the SBT are decided by the payload and the latency deadline.

Before three steps of the scheduling operation, a DRL-based algorithm is used to learn an optimal decision for scheduling request control. The RBs are allocated to slice requests that are successfully accepted and the allocation is maintained during the allocation window spanning multiple time slots. After receiving a slice request,

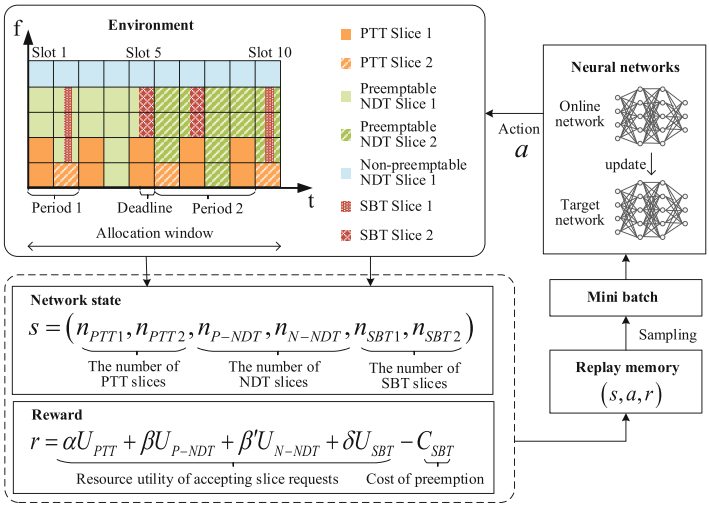


Fig. 5.7 The illustration for DRL-based resource scheduling

the slice manager needs to determine whether to allocate resources to this request. In order to maximize the resource benefit of the wireless TSN BS and use limited resources to realize more slices, a DRL model is adopted with its action space, state space and reward. As depicted in Fig. 5.7, the optimal actions are determined by deep neural network in response to the observed state and a reward is received for taking the good or bad action. The expected cumulative reward (i.e., Q-value) is used to incorporate farsighted system evolution into the decision-making and the decision strategies are updated through the feedback of previous decisions. Experience replay memory stores and samples historical rewards, actions, and state transitions into mini-batches, improving the training performance.

The state space consists of the numbers of different types of PTT, NDT and SBT slices which are successfully accepted in the RAN, and it is denoted as $\mathcal{S}=\{s_t\}, s_t = (n_{PTT}, n_{P-NDT}, n_{N-NDT}, n_{SBT})$, where s_t is the state at time t, n_{PTT} is the number of accepted PTT slices, n_{P-NDT} is the number of preemptable NDT slices, n_{N-NDT} is the number of unpreemptable NDT slices and n_{SBT} is the number of SBT slices. The number of slices belonging to the same type is counted for each state. With the increase of slice types, the state space becomes larger.

The action space denotes accepting/rejecting the incoming slice requests and the rejected requests need to wait for next allocation window, and it is represented by $\mathcal{A} = \{a_t\}, a_t = \{0, 1\}$, where a_t is the action at time t. $a_t = 0$ represents to reject the incoming slice requests while $a_t = 0$ represents to accept the incoming slice request.

The reward of accepting a slice request is defined as the utility U of accepting slice requests minus the preemption cost, which can be denoted as

$$r = \alpha \sum_{n=1}^{n_{PTT}} U_{PTT}^{n} + \beta \sum_{n=1}^{n_{P-NDT}} U_{P-NDT}^{n} + \beta' \sum_{n=1}^{n_{N-NDT}} U_{N-NDT}^{n} + \delta \sum_{n=1}^{n_{SBT}} U_{SBT}^{n} - C_{SBT}. \tag{5.8}$$

α, β, β', and δ are proportional to the priority of the slices, and C indicates the rate loss caused by resource preemption of SBT slices to users of preemptable NDT slices. The utility functions are defined particularly for different types of slices based on the priority [23]. Caused by resources preemption of SBT at the mini-slot timescale, there will be some possible loss of rate to NDT. Hence, the preemption cost is represented by the rate loss of the NDT users, which is a convex function of the fraction of preempted resources.

There are three typical models for the rate loss associated with resource preemption [2].

- **(1) Linear model:** The rate loss of the preemptable NDT is directly proportional to the fraction of preempted resources.
- **(2) Convex model:** The rate loss is modeled through a convex function of the fraction of preempted resources.
- **(3) Threshold model:** The NDT users are unaffected by resource preemption until a threshold and they suffers complete loss when the threshold is exceeded.

Here the similar convex model in [2] is utilized, the rate loss function of NDT user u when the channel state is in state s can be denoted by $h_u^s(x) = e^{k_u(x-1)}$, where k_u determines the sensitivity of an NDT user to the SBT user. $x = \frac{l_u^s}{m_u^s}$, where m_u^s denotes the amount of resources allocated to the NDT user u and l_u^s denotes the amount of resource that was preempted by the SBT from user u. By embedding the rate loss function in the formulation of reward, the proposed scheme achieves a trade-off between reducing rate loss for the preempted NDT and guaranteeing the low latency for the SBT.

The intelligent algorithms for the resource scheduling and configuration operations are provided by the AI-engine. Except for the conventional DRL algorithm, the A3C, DDPG, MADRL algorithms mentioned in Chap. 4 and other advanced DRL algorithms can also be encapsulated in the AI-engine, supporting intelligent slicing for wireless TSN BS.

Acknowledgments If you want to include acknowledgments of assistance and the like at the end of an individual chapter please use the `acknowledgement` environment—it will automatically render Springer's preferred layout.

References

1. 3GPP: Study on communication for automation in vertical domains (2018)
2. Anand, A., de Veciana, G., Shakkottai, S.: Joint scheduling of URLLC and eMBB traffic in 5G wireless networks. IEEE/ACM Trans. Networking **28**(2), 477–490 (2020). https://doi.org/10.1109/TNET.2020.2968373
3. Bakri, S., Frangoudis, P.A., Ksentini, A., Bouaziz, M.: Data-driven RAN slicing mechanisms for 5G and beyond. IEEE Trans. Netw. Serv. Manage. **18**(4), 4654–4668 (2021). https://doi.org/10.1109/TNSM.2021.3098193
4. Bhattacharjee, S., Katsalis, K., Arouk, O., Schmidt, R., Wang, T., An, X., Bauschert, T., Nikaein, N.: Network slicing for TSN-based transport networks. IEEE Access **9**, 62788–62809 (2021). https://doi.org/10.1109/ACCESS.2021.3074802
5. Bjϕmstad, S., Chen, D., Veisllari, R.: Handling delay in 5G ethernet mobile fronthaul networks. In: 2018 European Conference on Networks and Communications (EuCNC), pp. 1–9. IEEE, Piscataway (2018)
6. Cheng, X., Wu, Y., Min, G., Zomaya, A.Y.: Network function virtualization in dynamic networks: a stochastic perspective. IEEE J. Sel. Areas Commun. **36**(10), 2218–2232 (2018). https://doi.org/10.1109/JSAC.2018.2869958
7. Di Renzo, M., Zappone, A., Debbah, M., Alouini, M.S., Yuen, C., de Rosny, J., Tretyakov, S.: Smart radio environments empowered by reconfigurable intelligent surfaces: how it works, state of research, and the road ahead. IEEE J. Sel. Areas Commun. **38**(11), 2450–2525 (2020). https://doi.org/10.1109/JSAC.2020.3007211
8. Filali, A., Mlika, Z., Cherkaoui, S., Kobbane, A.: Dynamic SDN-based radio access network slicing with deep reinforcement learning for URLLC and eMBB services. IEEE Trans. Network Sci. Eng. **9**(4), 2174–2187 (2022)
9. Finn, N., Thubert, P., Varga, B., Farkas, J.: Deterministic networking architecture. In: RFC 8655. IETF, Fremont (2019)
10. Ginthör, D., Guillaume, R., Schüngel, M., Schotten, H.D.: 5G RAN slicing for deterministic traffic. In: 2021 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2021). https://doi.org/10.1109/WCNC49053.2021.9417296
11. Giordani, M., Polese, M., Mezzavilla, M., Rangan, S., Zorzi, M.: Toward 6G networks: use cases and technologies. IEEE Commun. Mag. **58**(3), 55–61 (2020). https://doi.org/10.1109/MCOM.001.1900411
12. Guan, W., Zhang, H., Leung, V.C.M.: Customized slicing for 6G: enforcing artificial intelligence on resource management. IEEE Network **35**(5), 264–271 (2021). https://doi.org/10.1109/MNET.011.2000644
13. Hermeto, R.T., Gallais, A., Theoleyre, F.: Scheduling for IEEE802. 15.4-TSCH and slow channel hopping mac in low power industrial wireless networks: a survey. Comput. Commun. **114**, 84–105 (2017)
14. Kuo, P.H., Mourad, A.: Millimeter wave for 5G mobile fronthaul and backhaul. In: 2017 European Conference on Networks and Communications (EuCNC), pp. 1–5. IEEE, Piscataway (2017)
15. Li, J., Liang, W., Xu, W., Xu, Z., Jia, X., Zhou, W., Zhao, J.: Maximizing user service satisfaction for delay-sensitive iot applications in edge computing. IEEE Trans. Parallel Distrib. Syst. **33**(5), 1199–1212 (2022). https://doi.org/10.1109/TPDS.2021.3107137
16. Lin, X., Li, J., Baldemair, R., Cheng, J.F.T., Parkvall, S., Larsson, D.C., Koorapaty, H., Frenne, M., Falahati, S., Grovlen, A., Werner, K.: 5G new radio: unveiling the essentials of the next generation wireless access technology. IEEE Commun. Stand. Mag. **3**(3), 30–37 (2019). https://doi.org/10.1109/MCOMSTD.001.1800036
17. Lo Bello, L., Steiner, W.: A perspective on IEEE time-sensitive networking for industrial communication and automation systems. Proc. IEEE **107**(6), 1094–1120 (2019). https://doi.org/10.1109/JPROC.2019.2905334

18. Nagata, S., Wang, L.H., Takeda, K.: Industry perspectives. IEEE Wireless Commun. **24**(3), 2–4 (2017). https://doi.org/10.1109/MWC.2017.7955902
19. Nasrallah, A., Thyagaturu, A.S., Alharbi, Z., Wang, C., Shao, X., Reisslein, M., ElBakoury, H.: Ultra-Low Latency (ULL) Networks: the IEEE TSN and IETF DetNet standards and related 5G ULL research. IEEE Commun. Surv. Tutorials **21**(1), 88–145 (2019). https://doi.org/10.1109/COMST.2018.2869350
20. Nixon, M., Rock, T.R.: A comparison of wirelesshart and ISA100. 11a. Whitepaper, Emerson Process Management, pp. 1–36 (2012)
21. Pop, P., Raagaard, M.L., Gutierrez, M., Steiner, W.: Enabling fog computing for industrial automation through time-Sensitive networking (TSN). IEEE Commun. Stand. Mag. **2**(2), 55–61 (2018)
22. Qiu, T., Chi, J., Zhou, X., Ning, Z., Atiquzzaman, M., Wu, D.O.: Edge computing in industrial internet of things: architecture, advances and challenges. IEEE Commun. Surv. Tutorials **22**(4), 2462–2488 (2020). https://doi.org/10.1109/COMST.2020.3009103
23. Schmidt, R., Chang, C.Y., Nikaein, N.: Slice scheduling with QoS-guarantee towards 5G. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–7 (2019)
24. Seijo, O., Iturbe, X., Val, I.: Tackling the challenges of the integration of wired and wireless TSN with a technology proof-of-concept. IEEE Trans. Ind. Inf. **18**(10), 7361–7372 (2022). https://doi.org/10.1109/TII.2021.3131865
25. Shen, X., Gao, J., Wu, W., Lyu, K., Li, M., Zhuang, W., Li, X., Rao, J.: AI-assisted network-slicing based next-generation wireless networks. IEEE Open J. Veh. Technol. **1**, 45–66 (2020)
26. Vlk, M., Hanzálek, Z., Brejchová, K., Tang, S., Bhattacharjee, S., Fu, S.: Enhancing schedulability and throughput of time-triggered traffic in IEEE 802.1Qbv time-sensitive networks. IEEE Trans. Commun. **68**(11), 7023–7038 (2020)
27. Wan, Z., Gao, Z., Di Renzo, M., Hanzo, L.: The road to industry 4.0 and beyond: a communications-, information-, and operation technology collaboration perspective. IEEE Network **36**(6), 157–164 (2022). https://doi.org/10.1109/MNET.008.2100484
28. Wang, G., Feng, G., Quek, T.Q.S., Qin, S., Wen, R., Tan, W.: Reconfiguration in network slicing—optimizing the profit and performance. IEEE Trans. Netw. Serv. Manage. **16**(2), 591–605 (2019). https://doi.org/10.1109/TNSM.2019.2899609
29. WhitePaper: industrial 5G edge computing: use cases, architecture and deployment. Technical report, 5G Alliance for Connected Industries and Automation (2022)
30. Wu, Y., Dai, H.N., Wang, H., Xiong, Z., Guo, S.: A survey of intelligent network slicing management for industrial IoT: integrated approaches for smart transportation, smart energy, and smart factory. IEEE Commun. Surv. Tutorials **24**(2), 1175–1211 (2022). https://doi.org/10.1109/COMST.2022.3158270
31. Yan, M., Feng, G., Zhou, J., Sun, Y., Liang, Y.C.: Intelligent resource scheduling for 5G radio access network slicing. IEEE Trans. Veh. Technol. **68**(8), 7691–7703 (2019). https://doi.org/10.1109/TVT.2019.2922668
32. Zhang, H., Guan, W., Wang, D., Song, Q., Nallanathan, A.: Demo: AI-engine enabled intelligent management in B5G/6G networks. In: 2022 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1–2 (2022)

Chapter 6
Conclusion and Future Works

6.1 Conclusions

As the development of 5G technologies enters a new stage, more and more vertical industries are deploying 5G systems to drive the application of emerging services. With the maturity of network slicing technologies, the differentiated demands of these emerging services can be satisfied by creating different type of slices on a common infrastructure network, reducing CAPEX/OPEX costs and improving resource utility. However, the addition of vertical industries brings challenges to multi-tenant network slicing. The complexity of heterogeneous resource management and E2E slices orchestration is increasing. Moreover, in most verticals, such as industrial Internet of Things scenarios, the number of ultra-low-latency and high-reliability service is gradually increasing, which requires customized resource allocation of network slices. To follow the trend of wireless networks from the consumer Internet to the industrial Internet, this book begins with the enabling technologies of network slicing, creates a new mode of cooperation between infrastructure networks for multi-tenant slicing, takes advantages of AI technologies, realizes the rapid deployment and orchestration of end-to-end slicing, enables intelligent control to avoid the service quality degradation of slices, and further proposes a customized slice management scheme which is suitable for industrial scenarios.

The main work of this book is summarized as follows:

1. **Efficient management of physical infrastructure networks.**
   Evolving from network sharing principles, mechanisms, and architectures to future on-demand multi-tenant systems, 5G networks need to provide resources for network slicing requests from multiple tenants. Therefore, the cooperation between multiple infrastructure networks with different structures needs to meet the requirements of multi-tenant slicing. However, the existing cooperation and sharing between infrastructure networks is only applicable to the scenario that data traffic request is peak. With the development of virtualization plat-

W. Guan, H. Zhang, *Network Slicing for Future Wireless Communication*, Wireless Networks, https://doi.org/10.1007/978-3-031-58229-5_6

form and cloud service platform, resource pooling technology will promote in-depth cooperation and sharing among infrastructure networks. Based on this, this chapter adopts the complex network theory to analyze topological characteristics of different infrastructure network and then propose a slice demand-oriented cooperation strategy between infrastructure networks based on two-side matching. With this cooperation strategy, a federated infrastructure network is formed to provide heterogeneous resources for cross-domain slices. In order to realize efficient resource allocation for these slices, the interconnections among infrastructure networks and the deployment mappings between slices and infrastructure networks are depicted by a multilayer model. Using this model, the management model of multi-domain slices is introduced, covering the whole process from receiving slice requests and releasing slice resources at the end of the slice life cycle. Following this model, the federated management framework of multi-domain infrastructure is described, supporting the cross-domain deployment of slices from different tenants in multiple domains of the federated infrastructure network. The management methods for multi-tenant slicing in this chapter are the basis of the subsequent works on slice deployment, slice enhancement, and slice for industrial applications.

2. **Intelligent deployment and orchestration of E2E slices.**
Realizing multi-tenant slicing requires to deploy E2E slices rapidly on the federated infrastructure network, satisfying the differentiated requirements of slices on resources of different domains. The existing researches about the deployment and orchestration of slices mainly focus on core network slices or RAN slices separately, lacking the fast deployment strategies for E2E slices. Moreover, the dynamic change of service requirements are hard to be handled, which results in the widely application of AI technologies in network slicing. However, adopting AI in the life-cycle management of multi-domain slices is still challenging. This chapter first provides a service-oriented E2E slice deployment policy, supporting dedicated deployment for three typical slices according to their characteristics of service requirements. Two stages of deploying slices on infrastructure, VNF placements and chaining VNFs, are optimized for different types of slices with exclusive optimization goals, achieving better resource efficiency and higher revenue of service provision. In addition to the rapid deployment strategy of E2E slices, this chapter also introduces an AI-based hierarchical resource management framework. On the one hand, this framework controls the admission of slice requests from different tenants through a global perspective, and on the other hand, it conducts local adjustments to the accepted slice requests as needed. Leveraging the DRL-based algorithms, this framework enables customized resource utilization for multi-tenant slicing by global resource allocation and local slice adaption. Following this framework, the AI-enabled slice orchestration mechanism is also illustrated in this chapter, improving resource efficiency while maintaining service quality for admitted slices. Moreover, an advanced DRL-based slice reconfiguration method is also introduced with the aim of realizing slice adaption at the lowest cost.

3. **AI-based performance enhancement for multi-tenant slicing.**
   As more and more tenants join in the business model of network sharing, the number and types of slices deployed on the same federated infrastructure network increase. The resource competition among slices will affect the transmission performance of slices and threaten the quality of service. Therefore, it is required to analyze the relationship between transmission performance and resource allocation of slices, and enhancing the performance of data traffic transmission on slices through elastic resource management strategy is urgent. Using some advanced AI methods in performance enhancement for multi-tenant slicing is becoming a trend. In this chapter, the new collaboration business model for multi-tenant slicing is presented and the problems existed in this model are listed. In order to tackle the problems caused by the resource competition among slices and realize dynamic QoS provision for multi-tenant slicing, traffic performance analysis of multiple isolated slices is performed. The multilayer complex network model is established, providing a clear global perspective for service providers to analyze the influence of the resource competition. Based on this multilayer model, data traffic of slices deployed in the same infrastructure network is also modeled, the effect of resource allocation parameters, the nodal coverage of slices and the number of slices on traffic performance is analyzed. Based on the analysis results, two efficient control strategies for avoiding resource competition among slices are proposed. Selectively increasing the resources of a few hub nodes in the infrastructure network or increasing the scale of the infrastructure network can effectively alleviate the congestion on the slices and improve the transmission performance. To better enforce these control strategies, three promising DRL algorithms, A3C DRL, DDPG, and MADRL, are introduced here.
4. **Customized slicing for industrial applications.**
   As 5G enters the phase 2, services which require ultra-low delay and high reliability will be brought to wireless networks. New requirements of emerging applications in vertical industries, such as smart manufacturing in Industry 4.0, draw higher demand in 5G techniques. A growing number of industries are already deploying 5G systems in the industrial scenarios, creating new revenue opportunities and promoting the automation of industrial manufacturing. Since the 5G standards specified in 3GPP has added functions to support the integration of 5G and TSN, it is necessary to develop an appropriate approach of network slicing for 5G TSN integration. To realize customized slicing for 5G TSN integration, this chapter first introduces some emerging industrial use cases for Industry 4.0 and analyzes the requirements of these use cases. According to the characteristics of these use cases, the data traffic types of individual applications are classified into three categories, sporadic burst traffic, periodic time-sensitive traffic, and non-deterministic traffic. Then, the work on standardization within IEEE 802.1 TG is summarized, including the clock synchronization in IEEE 802.1AS/ASrev, frame preemption in IEEE 802.1Qbu-2016, time-triggered (TT) communication in IEEE 802.1Qbv-2015, as well as traffic filtering and policing

in IEEE P802.1Qci-2017. Also the work of DetNet WG which includes flow management and flow integrity is reviewed. Besides these efforts on deterministic communication, researches on the QoS-aware traffic scheduling toward 5GS as a logical TSN bridge also provide support for deterministic traffic transmission in 5GS. To enable the simultaneous transmission of time-critical and nontime-critical traffic in 5GS as a logical TSN bridge, an AI-based time-sensitive RAN slicing framework is proposed in this chapter, deploying AI-engine in the fog node. AI algorithms encapsulated in the AI-engine are adopted, achieving intelligent resource slicing in 5G-TSN integration.

6.2 Future Work

With the development of wireless networks, the application scenarios in the future will be more diverse and complex. The demand for efficient utilization of network resources, user experience, and security guarantee will make the future work of wireless network slicing continue to develop and improve in the following several directions.

1. **Automative and intelligent management.** In the future, the complexity of wireless network service requirements increases and users demand interactive experience, which requires more precise network resource allocation and management. Although there are already some AI-based resource management frameworks for network slicing [2], the relatively long convergence time of ML methods undermines their usefulness. Besides convergence, the stochastic nature of the wireless network may require ongoing updates of the parameters and continuous adaption of ML methods. Therefore, developing feasible and scalable ML algorithms in automative and intelligent network management need more study and analysis.
2. **Flexible and adaptive slicing.** Future wireless network slicing technology will evolve toward adaptability [3], where the configurations of cross-domain slices can be adjusted flexibly. The goals of resource slicing are still satisfying various service requirements and achieving efficient utilization of network resources in a real-time way. As user demands and network conditions are highly dynamic and uncertain, there are some efforts in the current literature to predict users' behavior. However, correlating the evolutional tendency of demands to resource allocation in network slicing constitutes a challenging line of research, where the complexity of real-time slice adaption cannot be neglected.
3. **Security and privacy protection.** With the widespread deployment of 5G networks in vertical industries, security and privacy protection become critical issues [1]. The future development direction will pay more attention to isolation among slices and secure communication, safeguarding the security of user data and privacy. It is essential to protect sensitive data and privacy information during the life cycle of network slicing, execute security updates and vulnerability fixes

to multimodal devices in a timely manner. It is important to filter and check the type of data traffic based on AI technologies to prevent malicious traffic from entering the slice.

References

1. Chahbar, M., Diaz, G., Dandoush, A., Cérin, C., Ghoumid, K.: A comprehensive survey on the E2E 5G network slicing model. IEEE Trans. Netw. Serv. Manag. **18**(1), 49–62 (2021). https://doi.org/10.1109/TNSM.2020.3044626
2. Habibi, M.A., Han, B., Fellan, A., Jiang, W., Sánchez, A.G., Pavon, I.L., Boubendir, A., Schotten, H.D.: Toward an open, intelligent, and end-to-end architectural framework for network slicing in 6G communication systems. IEEE Open J. Commun. Soc. **4**, 1615–1658 (2023). https://doi.org/10.1109/OJCOMS.2023.3294445
3. Wei, F., Feng, G., Sun, Y., Wang, Y., Qin, S., Liang, Y.C.: Network slice reconfiguration by exploiting deep reinforcement learning with large action space. IEEE Trans. Netw. Serv. Manag. **17**(4), 2197–2211 (2020). https://doi.org/10.1109/TNSM.2020.3019248