Chapter 7
Digital Twin for Aerial-Ground Networks

Abstract With the widespread deployment of unmanned aerial vehicles (UAVs) in civil and military fields, researchers have turned their attention towards the emerging area of aerial-ground networks for computing-intensive applications, data-intensive applications, and network-intensive applications. However, the application of aerial-ground networks relies on dynamic perceptions and intelligent decision making, which are difficult to conceive of due to the heterogeneity of ground devices and the complexity of the aerial-ground environment. The convergence of digital twin (DT) and UAVs has great potential to tackle the challenge and improve the service quality and stability in applications such as rescue and search and communication relaying. This chapter first investigates the advantages, challenges, and key techniques of DTs for aerial-ground networks. In addition, we highlight the main issues of DT for UAV-assisted aerial-ground networks with two case studies, including cross-domain resource management and intelligent cooperation among devices.

7.1 Introduction

Recently, aerial-ground networks based on unmanned aerial vehicles (UAVs) have made great success in various applications, such as disaster relief, service congestion, and damage assessment. Thanks to their inherent advantages, such as wide coverage, high flexibility, and strong resilience, UAVs can act as aerial mobile base stations to provide seamless and intelligent services for ground devices. However, due to the heterogeneity and mobility of ground devices and the dynamic network topology, the advantages of aerial–ground networks cannot be fully exploited. As an emerging digital mapping technology, digital twin (DT) has great potential to tackle the network dynamics and complexity of aerial-ground networks. By mapping the channel state and computing state, DT established on UAV can reflect the state of ground devices or network topology in a timely manner and accurately capture their state changes. After learning from these complex statuses, DT established on UAVs can support diversified applications, such as trajectory planning, large-scale mapping, urban

Y. Zhang, *Digital Twin*, Simula SpringerBriefs on Computing 16,
https://doi.org/10.1007/978-3-031-51819-5_7

modelling, road patrol, and anti-piracy. We detail the main application scenarios of DT deployed on UAVs in different fields as follows.

- *Smart city*: In smart cities, with the help of DT deployed on UAVs, we can build a large-scale virtual city, dynamically monitor urban facilities, allocate urban public resources, and further realize intelligent collaborative decision making in urban management.
- *Disaster rescue*: In the disaster rescue field, UAVs with DT can analyse the connection performance of ground rescue devices and then make proactive communication resource allocations for high-priority devices to maximize the long-term quality of service (QoS).
- *Telemedicine*: In the field of telemedicine, through smart wearable devices, patients' health information can be sent back to DTs deployed on UAVs. UAVs with DT can track and monitor a patient's health status remotely and in a timely manner. When the DT measures any abnormal information, the rescue agency can immediately provide first aid services.
- *Internet of vehicles*: In the Internet of Vehicles (IoV), UAVs with DT can competently implement the real-time planning of vehicle trajectories in a specific area. At the same time, services such as status awareness and mobility prediction provided by DT can effectively avoid traffic congestion and reduce traffic accidents.

DT is able to assist in the optimal allocation and intelligent dispatching of valuable aerial resources. We further summarize the advantages of DT and UAV fusion as follows.

- *Hyperconnectivity*: Due to the wide coverage of UAVs over ground devices, DT deployed on UAVs can achieve interoperability and hyperconnectivity with physical counterpart devices. DT deployed on a UAV can connect all the ground devices in the aerial-ground network. We can fully utilize the advantages of DT from the multidimensional integration of information to sense how different devices work together, thus building an aerial-ground network with hyperconnectivity. In an aerial-ground network with hyperconnectivity, DT has the interaction details and status information of all the devices, which can then dynamically provide optimal decisions for different problems.
- *Low latency*: Thanks to the mobility of UAVs, DT deployed on a UAV can maintain a specified synchronization frequency with the ground device, which enhances the fidelity of the signal and brings more reliable DT services to the device. DT is sensitive to synchronization frequency, and untimely state synchronization or instruction updates can cause DT to make incorrect decisions. UAVs have the ability to move with mobile devices such as vehicles, significantly reducing DT status update delays due to communication distance. DT deployed on UAVs can better meet the requirements of devices for low network latency in different application scenarios, such as real-time trajectory planning in IoV, and provide services with higher performance and reliability.
- *Strong stability*: DT deployed on UAVs can monitor the status of each aerial-ground network in real time, which ensures the coverage and stability of the

aerial-ground network and makes the DT service more stable. Deploying DT on the ground makes it difficult to perform timely maintenance in the event of an attack or communication failure, which will lead to the interruption of DT services. DT deployed on a UAV can closely monitor the state changes in different aerial-ground networks, after detecting emergency situations such as UAV damage and network failure. DT can then immediately replenish and replace UAVs, continuously providing high-stability and high-performance services for devices.

The complementary advantages of DT and UAV play an important role in diverse applications that require stable network connections. However, there are still challenges in how to customize DT on UAVs for smart services in aerial-ground networks. We summarize the challenge in two cases.

- *Cross-domain resource allocation*: An aerial-ground network involves two different resource domains: the aerial domain and the terrestrial domain. The main challenge that DT faces on UAVs is the effective allocation of limited resources across domains under resource and distance constraints. DT-enabled intelligent services are often supported by a large amount of data distributed over various terminal devices. In a large-scale aerial-ground network, there are limitations of physical distance, communication resources, and computing resources; therefore, how DT deployed on UAVs effectively allocates resources across domains deserves in-depth study. In addition, the limited energy capacity of UAVs cannot support DT modelling, and DT relies on abundant computing resources and sufficient energy supply, which further limits the endurance of UAVs.
- *Cross-device intelligent collaboration*: The intelligent collaboration of different devices in an aerial-ground network is an important link to keep the network running efficiently. One of the important features of aerial-ground networks is a highly dynamic network environment. Diverse devices are constantly joining and withdrawing from the network, and mobile devices such as vehicles, UAVs, and mobile phones have low latency tolerance. For DTs deployed on UAVs, enabling different devices to achieve dynamic joint decision making and intelligent collaboration in tasks such as autonomous driving and trajectory planning while reducing network latency is challenging.

7.2 Key Techniques

7.2.1 Cross-Domain Resource Management

Aerial-ground networks can enhance the environmental perception and decision making capabilities of the network by leveraging multidimensional resources to achieve resource management. However, the resources in different domains (such as air and ground) are complicatedly coupled, and the orchestration of these cross-domain resources is confronted with a huge state–action space, which makes it difficult to allocate resources optimally in real time [78, 79]. To effectively manage the multidimensional resources (for communication, computing, and caching) of aerial-ground networks, the state change and QoS of the network are the key factors to consider.

Ensuring the flexibility and efficiency of resource management:

Aerial-ground networks are extremely dynamic and complex because of the high mobility of heterogeneous devices and the large scale of the networks. It is difficult to achieve flexible and efficient network resource management. As an emerging digital mapping technology, DT provides an approach for realizing effective and reliable network orchestration by mapping and predicting the dynamics of networks. Deng *et al.* in [80] proposed a combined approach of expert knowledge, reinforcement learning, and DT to cope with the dynamic changes of high-dimensional network states. Dai *et al.* in [74] proposed a new paradigm DT network for the Industrial Internet of Things (IIoT) and formulated random computing shunting and resource allocation problems, using Lyapunov optimization technology to transform the original problem into a deterministic per-slot problem. Lu *et al.* in [37] proposed a DT edge network to fill the gap between the physical edge network and the digital system. The integration of DT technology into aerial-ground networks can yield considerable improvement in both the latency performance and computing efficiency of applications running on ground devices and aerial devices.

Software-defined networking (SDN) can be utilized to construct and manage virtual networks to support specific network services for flexible network management [81]. Based on SDN architecture, Li *et al.* in [82] modelled multidimensional resource scheduling as a partially observable Markov decision process and used value iteration to jointly optimize networking, caching, and computing. Due to the complicated coupling of multidimensional resources, the central controller can hardly know a priori the effects of its actions on system performance. To this end, He *et al.* in [83] proposed a resource orchestration method based on deep reinforcement learning, with which the central controller learns an effective policy via trial-and-error search.

Ensuring QoS performance: The effective management of the multidimensional resources (for communication, computing, and caching) of aerial-ground networks to guarantee the required QoS performance of ground devices is also an important challenge. High computational complexity, the large cost of equipment deploymemt, and limited resources are the factors that hinder the improvement of QoS performance.

- *Reducing computational complexity*: Due to the limited computing and communication capabilities of ground devices, task offloading, as a key technology, can effectively improve service execution efficiency and realize the fast and efficient response of ground devices. Task offloading means that resource-constrained mobile terminal devices can offload overloaded computing tasks to edge nodes with stronger computing or communication capabilities, to improve computing speed and save energy. For example, road side units (RSUs) can undertake computation-intensive tasks (e.g. semantic image segmentation, motion planning, and route planning) for vehicles. Xu et al. in [30] proposed a service offloading method with deep reinforcement learning in DT-empowered IoV to provide vehicular services with a high QoS level. To reduce processing delays, Do-Duy et al. in [84] proposed a novel DT framework assisting in the task offloading of IoT devices for IIoT networks with mobile edge computing. Qu et al. in [85] proposed a deep meta-reinforcement learning offloading algorithm that combines multiple parallel deep neural networks with Q-learning, quickly and flexibly obtaining the optimal offloading strategy from a dynamic environment.

- *Reducing equipment deployment costs*: To achieve effective resource management and satisfy the diverse QoS requirements, the deployment cost of edge nodes cannot be ignored in aerial-ground networks. Using a large number of edge nodes to completely cover an area means a large deployment cost. When ground devices offload computing tasks to nearby edge nodes through the assistance of UAVs, the appropriate incentive is required for edge nodes to contribute their services. Edge nodes can be unwilling to contribute their services if the rewards cannot compensate for their service costs. Sun et al. in [86] designed an incentive mechanism to motivate RSUs to provide computing resources for ground vehicles. It was able to effectively complete vehicle task offloading schemes with the assistance of UAVs in an aerial-ground network. Zhou *et al.* in [87] proposed a novel incentive-driven and deep Q-network–based method and combined a content caching strategy and incentive mechanism to improve the performance of device-to-device offloading. To realize the long-term stability of DT services, Lin et al. in [88] designed an incentive-based congestion control scheme to offload real-time mobile data captured by DT to mobile edge computing servers.
- *Reducing the burden of aerial devices with limited resources*: Most works ignore the fact that centralized resource allocation schemes introduce a great burden to aerial devices, especially to UAVs in aerial-ground networks. Moreover, the incentive mechanism can be computation intensive, which results in service-unrelated energy consumption and further deteriorates service endurance. Thus, the resource allocation scheme should be carried out in a distributed manner. Through cooperative networks [89], SDN controllers can be decomposed into multiple simpler controllers to reduce the complexity of a large action space. Nasir *et al.* in [90] thus leveraged multi-agent deep Q-learning to distributedly schedule power allocation in wireless networks. The alternating direction method of multipliers (ADMM) is a distributed parallel optimization algorithm, and resource allocation problems based on ADMM have attracted much attention. Wang *et al.* in [91] considered computational offloading, resource allocation, and content caching strategies as optimization problems. An algorithm for solving optimization problems based on the ADMM algorithm was designed. Liang *et al.* in [92] proposed an efficient ADMM-based distributed virtual resource allocation algorithm in virtualized wireless networks. In addition, Zheng *et al.* in [93] designed a converged and scalable Stackelberg game–based ADMM for edge caching to solve storage allocation games and user allocation games in a distributed manner.

7.2.2 Cross-Device Intelligent Cooperation

In aerial-ground networks, heterogeneous ground devices can collaborate with aerial devices to accomplish intelligent network orchestration based on federated learning. Cross-device intelligent cooperation plays an important role in the efficient operation of networks and stable network environments. However, due to the heterogeneity, mobility, and selfishness of devices, across-device intelligent cooperation based on federated learning still faces many challenges. For example, further optimization is needed in terms of communication efficiency, training efficiency, and training costs. DT has the powerful ability to capture the state of heterogeneous devices in real time, which can effectively promote cross-device intelligent cooperation.

Improving communication efficiency: The heterogeneity and high mobility of devices complicate network management. The real-time changes of device states can lead to inaccurate channel estimation and affect the communication efficiency of federated learning. DT can analyse the connection performance of devices and make proactive communication resource allocations for improving communication efficiency. Lu *et al.* in [9] proposed a blockchain-based DT-enabled federated learning scheme to improve communication efficiency. Tran *et al.* in [94] studied the collaborative optimization problem when devices participate in federated learning in wireless networks. By adjusting a device's resource allocation strategy and the local training update frequency between two global aggregations, the best trade-off between communication time and computing performance can be achieved. Sun *et al.* in [33] used deep reinforcement learning to adaptively adjust the cooperative aggregation strategy of federated learning to achieve the balanced optimization of communication and computing. Krouka *et al.* in [95] proposed a novel distributed reinforcement learning algorithm to solve the random interference and communication interference of wireless channels and optimize communication efficiency.

Improving training efficiency: The dynamic nature of aerial-ground networks makes it difficult for heterogeneous devices to complete collaborative computing, so it is difficult to improve the training efficiency of federated learning. Lu *et al.* in [37] proposed a blockchain-empowered federated learning framework operating in a DT wireless network that comprehensively considers DT association, training data batchsize, and bandwidth allocation to formulate the training optimization problem. Jiang *et al.* in [36] exploited blockchain to propose a new DT edge network framework and designed a joint cooperative federated learning and local model update verification scheme that achieves the optimal unified time. Zhang *et al.* in [96] proposed a reinforcement of a federated learning scheme based on deep multi-agent reinforcement learning to optimize the training performance of federated learning in distributed IIoT networks. Li *et al.* in [97] proposed a platform-assisted collaborative learning framework. This framework can rapidly adapt to learning a new task at the target edge node by using a federated meta-learning approach with a few samples. Existing collaborative computing needs to restart learning as the topology changes, which leads to the failure or slow convergence of the established cooperative mechanism. DT can capture a complex network topology dynamically and improve the efficiency of collaborative computing between devices.

Reducing training costs:

It is necessary to encourage heterogeneous devices to participate in intelligent cooperation. Heterogeneous devices need to spend their resources and costs to train the federated learning model. They are therefore reluctant to participate in training without appropriate incentives [98, 99]. Existing incentive mechanisms can perform poorly due to the insufficient utilization of massive data and inaccurate modelling of operations in dynamic aerial-ground networks. DT can reduce the information asymmetry between devices by monitoring the status of devices in real time. Yang *et al.* in [100] introduced the Stackelberg game to establish an interaction model that comprehensively considers the data size, training time, and power consumption to measure the contribution to motivate client participation. Lim *et al.* in [101] studied the incentive mechanism for federated learning in UAV-assisted IoV to encourage contributions from data owners, considering information asymmetry between UAVs and the data owners. Federated learning is data driven, and the motivation of clients and the quality of data they provide have an important impact on the training results [102, 103]. The incentive mechanism combined with DT is suitable for motivating heterogeneous devices to actively participate in training in aerial-ground networks.

In summary, cross-device intelligent cooperation based on federated learning in an aerial-ground network still needs to be studied further. The integration of DT and aerial-ground networks can provide favourable support for realizing the cooperation mechanism of model-free, self-learning, autonomous intelligence.

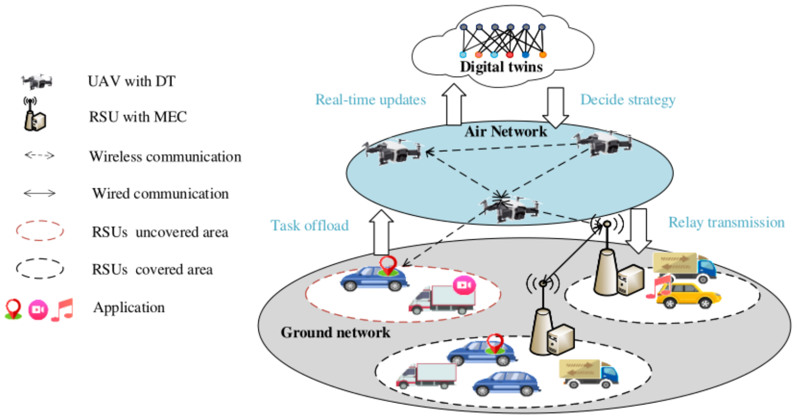

Fig. 7.1 A DT-driven aerial-ground network system model

7.3 DT for Task Offloading in Aerial-Ground Networks

7.3.1 System Model

To realize the efficient allocation of cross-domain resources from air and ground, we establish a dynamic DT model for an aerial-ground network. The DT model can capture the time-varying demand and supply of cross-domain resources in the network. Deploying the DT model on devices in the network can significantly improve the environmental perception, computing efficiency, and delay performance of the devices. This is beneficial for the unified and efficient resource allocation and scheduling in aerial-ground networks.

As shown in Fig. 7.1, we consider a DT-driven aerial-ground network in which vehicles act as ground devices and UAVs act as aerial devices. The network is composed of vehicles, RSUs, UAVs and DTs. We assume the UAVs are responsible for areas that are not covered by RSUs, as a supplement to the ground network. In such areas, vehicles and RSUs are able to deliver messages to a UAV directly with line-of-sight communication. With the assistance of UAVs, the vehicles not covered by the ground network could offload their computing tasks to RSUs to reduce their computing burden. We establish two DT models, including the DT of a group of RSUs and the DT of vehicles. Both DTs are established in UAVs to update the network topology and traffic load in real time and help UAVs make specific decisions, such as path planning. The DT of a group of RSUs can be given by

$$\mathcal{D}^r = \{\mathcal{F}^r, G^r, L^r\}, \tag{7.1}$$

where $\mathcal{F}^r$ is a vector describing the available computing resource status of the RSUs, G^r is the network topology between the RSUs, and L^r is the network transmission load of the RSUs.

The DT of vehicles can be given by

$$\mathcal{D}^v = \{G^v, L^v, C, Q\}, \tag{7.2}$$

where G^v is the network topology of the vehicles, L^v is the communication load of the vehicles, C represents the demand information of the vehicles at this time, and Q is the preference of the vehicles for the resource providers. The preference is determined by the historical service of the vehicles in a specific type of offloading task.

7.3.2 Utility Function

The DT of a group of vehicles and the DT of a group of RSUs have different utilities. The set of RSUs in the network is $\mathcal{M} = \{1, ..., m, ..., M\}$. The set of vehicles that require offloading tasks is $\mathcal{N} = \{1, ..., n, ..., N\}$. Vehicle n wants to maximize its service satisfaction, which is the accumulated satisfaction it achieves from various RSUs. A vehicle's satisfaction is defined as the ratio of its cumulative satisfaction from RSUs to the total number of resources it receives. Thus the satisfaction of vehicle n is given by

$$\mathcal{S}_n = \frac{\sum\limits_{m \in \mathcal{M}} \{q_{n,m} p_{m,n} - \frac{q_{n,m} p_{m,n}^2}{2\tilde{f}}\}}{\sum\limits_{m \in \mathcal{M}} p_{m,n}}, \tag{7.3}$$

where $\tilde{f}$ is the maximum expected value of resources from RSUs, $p_{m,n}$ represents the CPU frequency obtained by vehicle n at RSU m, and $q_{n,m}$ represents the preference of vehicle n for RSU m.

The DT of RSUs tries to minimize energy consumption. The energy consumption on the RSU is related to the frequency and duration of the CPU used. We can express energy consumption as

$$E(\mathcal{P}) = \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} \omega p_{m,n}^2 c_{n,m}, \tag{7.4}$$

where ω represents the effective capacitance parameter of the computing chipset, and $c_{n,m}$ is the number of CPU cycles required for RSU m to calculate its tasks for vehicle n. The detailed resource scheduling of each vehicle is expressed as $\mathcal{P} = \{P_m, m \in \mathcal{M}\}^T$.

7.3.3 Distributed Incentives for Satisfaction and Energy Efficiency Maximization

The goals of RSUs and the DT of RSUs are different. An RSU is designed to maximize the average satisfaction of the vehicle. whereas an RSU's DT is designed to maximize global energy efficiency. Although these quantities are used to formulate the allocation scheme of computing resources, it is difficult to achieve the goal of

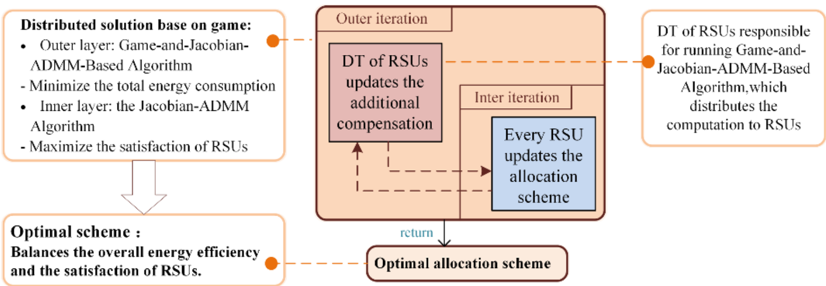

Fig. 7.2 Workflow of a game and Jacobian ADMM-based algorithm

minimizing total energy consumption when they have different optimal values. In addition, due to limited computing resources, computationally intensive centralized computing creates pressure for UAVs. Therefore, we propose an incentive mechanism based on the Stackelberg game and Jacobian ADMM to allocate computing resources, so that the DT of RSUs and the RSUs can reach a consensus on the allocation scheme and solve the whole problem in a distributed and parallel manner.

Due to the complexity of solving the desired objectives of RSUs and the DT of RSUs, we first derive the optimization problem of RSUs and the DT of RSUs and then construct a Stackelberg game. We solve the average satisfaction maximization problem for vehicles and the global energy efficiency maximization problems for the DT of RSUs by using the classic ADMM and Jacobian ADMM with two blocks, respectively. We obtain the resource allocation schemes of the two problems (the DT-driven classic ADMM and the DT-driven Jacobian ADMM). Furthermore, we model these two problems as a complete Stackelberg game. In the game, the RSUs' DT is the leader and the RSUs are the follower. According to the goals of RSUs and the DT of RSUs, we can formulate the Stackelberg game as

$$\begin{aligned} Leader: \quad & \underset{\mathcal{P}}{minimize} \quad E(\mathcal{P}) \\ Follower: \quad & \underset{p_m, \overline{Q}_m}{minimize} \quad \Phi_m(h_m(P_m, \eta_m), \boldsymbol{\theta}_m) \\ \text{s.t.} \quad & \sum_{n \in \mathcal{N}} p_{m,n} = f_m, m \in \mathcal{M} \quad (C1), \end{aligned} \tag{7.5}$$

where $\overline{Q}_m$ is the cumulative preference of all the vehicles for RSU m. The term $\Phi_m(\cdot)$ includes the optimization direction of the DT of the RSUs and RSU m and the compensation from the DT of the RSUs. The classic ADMM with two blocks is powerless in this kind of convex optimization problem with high-dimensional variables. The Jacobian ADMM-based algorithm is able to solve convex optimization problems by breaking them into smaller subproblems, making each part more tractable. Therefore, we use the game and Jacobian ADMM-based algorithm to solve the problem. The algorithm flow is shown in Fig. 7.2.

In the beginning, the DT of the RSUs, as the leader, sends the incentive parameter $\boldsymbol{\theta}_m$ to the corresponding RSU m, that is, the additional compensation of the DT of the RSUs to RSU m. We define the number of iterations of the outer loop as k. At iteration k, given incentive parameters $\{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_m, \cdots, \boldsymbol{\theta}_M\}$ from the leader, each RSU updates its own computing resource allocation scheme P_m in the inner loop, and then the leader and the follower can reach the current optimal scheme. At the next iteration, $k+1$, the leader will adjust the incentive parameters based on the updated $P_m, \forall m \in \mathcal{M}$. Then, a new current optimal scheme can be reached. When the outer iteration is terminated, the optimal incentive parameters and resource allocation scheme are the equilibrium point $(\boldsymbol{\theta}^*, \mathcal{P}^*)$ of the Stackelberg game. The proposed DT-driven game ADMM minimizes global energy consumption based on the premise of ensuring the satisfaction of the RSUs.

7.3.4 Illustration of the Results

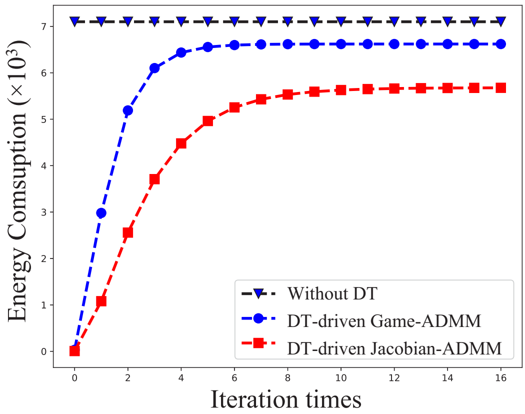

Fig. 7.3 Convergence of the energy consumption of all RSUs over iterations

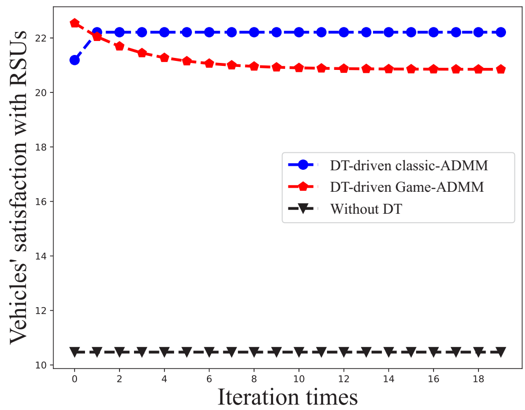

Fig. 7.4 Vehicle satisfaction with RSUs over iterations under three schemes

Figure 7.3 compares the energy consumption of three schemes, that is, the DT-driven Jacobian ADMM, the DT-driven game ADMM, and the scheme without DT, over the numbers of iterations. The scheme without DT allocates resources without the preference information that was obtained from the DT of the vehicles.

The energy consumption of the scheme without DT is the highest and remains a constant, because the tasks and CPU frequency can only be allocated randomly. This leads to a decision making and optimization process without iteration. The energy consumption of the DT-driven Jacobian ADMM is the lowest, since minimizing total energy consumption is its only objective at the cost of low vehicle satisfaction. The proposed DT-driven game ADMM jointly considers the overall energy efficiency and the satisfaction of the RSUs, and its energy consumption is thus higher than that of the DT-driven Jacobian ADMM.

Figure 7.4 compares the vehicles' satisfaction with the RSUs of three schemes, that is, the DT-driven classic ADMM, the DT-driven game ADMM, and the scheme without DT. Due to the contradictory goals of the RSUs and the DT of the RSUs, the DT-driven game ADMM attempts to balance between the two contradictory goals, and its satisfaction is a bit lower than that of classic ADMM. This is because RSUs allocate a great deal of resources to vehicles with high preferences, to provide satisfactory services for the vehicles. The satisfaction achieved by both DT-driven schemes, that is, the DT-driven Jacobian ADMM and the DT-driven game ADMM, is much higher than that without DT. This is because, in the scheme without DT, the preferences of the vehicles for RSUs are unknown, and the allocation cannot fully meet the actual requirements of the vehicles.

7.4 DT and Federated Learning for Aerial-Ground Networks

7.4.1 A DT Drone-Assisted Ground Network Model

Figure 7.5 shows a drone-assisted ground network scenario consisting of drones, ground clients, and DTs, where the drones provide supplementary capacity for ground communications during natural disasters or traffic peaks. Mobile drones with a wide range of coverage act as servers, responsible for task offloading, global model updates, and so forth. A wide variety of ground equipment, such as smartphones and laptops, serves as clients to perform tasks and connect with drones through wireless communications.

The drone serving as the aggregator cooperates with the ground equipment serving as the trainers to perform federated learning tasks. The drone publishes a global model ω, which all participating clients will download. Then, each client uses its own private data sets to train the model and upload the new weights or gradients to the server. This process is conducted iteratively until the entire training process converges [104, 74].

The establishment of DT can capture the state of network elements in real time and effectively help the system make intelligent decisions. DT types include the DT of ground clients and the DT of the drone. The DTs of ground clients are deployed on a resource-rich ground node. The drone would maintain the DT by exchanging information with the ground node instead of all the clients. The set of clients in the network is $\mathcal{N} = \{1, 2, \cdots, N\}$. Client i's DT, DT_i^c, at time t can be expressed as

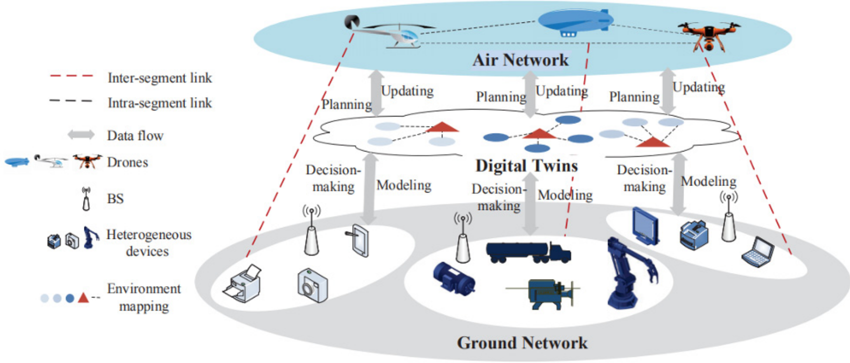

Fig. 7.5 The architecture of a DT-empowered aerial-ground network

$$DT_i^c(t) = \{F_i^t(\omega), b_i(t), f_i(t)\}, \tag{7.6}$$

where ω denotes the current training parameter of client i, $F_i^t(\omega)$ represents the current training state of client i, $b_i(t)$ represents the packet loss rate, and $f_i(t)$ is the CPU frequency of the client at time t.

Due to the deviation of DTs, the packet loss rate deviation $\hat{b}_i(t)$ and the CPU frequency deviation $\hat{f}_i(t)$ can be measured as the errors of the DT mapping in the communication environment and computing power, respectively. For client i, the calibrated DT is

$$\hat{DT}_i^c(t) = \{F_i^t(\omega), b_i(t) + \hat{b}_i(t), f_i(t) + \hat{f}_i(t)\}. \tag{7.7}$$

The DT of the drone manages the deviation of the DTs of the clients and has a preference for the clients. Drone j's model is

$$DT_j^u(t) = \{\mathcal{P}(t), \hat{\mathcal{D}}(t)\}, \tag{7.8}$$

where $\mathcal{P}(t)$ is the reputation distribution of nodes within its coverage area, and $\hat{\mathcal{D}}(t)$ is the set of deviations between the client's local update and the global update.

7.4.2 Contribution Measurement and Reputation Value Model

Update significance can intuitively measure the contribution of a local model update to the global model update. The update significance is measured by the model deviation d_i^τ, which is the divergence of a particular local model from the average across all local models. A small d_i^τ reflects a high quality of upload parameters of client i. The aggregator updates the value of d_i^τ for client i in each time slot, as a basis for the quality evaluation of the parameters submitted by client i.

The reputation of a client can also affect the training process. Through the reputation model, high-performance clients should be identified in terms of sufficient communication resources, powerful computing capabilities, and accurate training results. We use $\mathcal{P} = (\rho_1, \rho_2, \cdots, \rho_N)$ to represent the reputation value of each client. According to subjective logic, the reputation value model is related to the communication capability of node i during the τth global update and the learning quality d_i^{τ}.

7.4.3 Incentive for Federated Learning Utility Maximization

Static and dynamic incentives are designed for small-scale networks and large-scale networks, respectively [37]. In a small-scale network, a single drone can cover all the clients. Therefore, we first design a static incentive mechanism. The term τ_i represents the decision of client i, that is, the number of rounds in which the client participates in the global update; $\mathcal{T} = (\tau_1, \tau_2, \cdots, \tau_N)$ represents the strategies for all the clients; and $\tau_{-i} = (\tau_1, \cdots, \tau_{i-1}, \tau_{i+1}, \cdots, \tau_N)$ denotes the training strategies of all the clients except for client i. Given the computing cost per round (a complete global update round) $C = (c_1, c_2, \cdots, c_N)$ and the communication cost per round $\mathcal{K} = (k_1, k_2, \cdots, k_N)$, the static incentive utility function is the difference between the reward and loss of client i, which can be defined by

$$U_i(\tau_i, \tau_{-i}) = \frac{\rho_i \tau_i}{\sum\limits_{j \in N} \rho_j \tau_j} R - \tau_i c_i - \tau_i k_i. \tag{7.9}$$

The utility function of the aggregator is the total energy consumption of clients in the learning process minus the payment of the aggregator. The static incentive utility function is defined as

$$U_0(R) = \sum_{i \in N} \rho_i \tau_i c_i - \alpha R^2, \tag{7.10}$$

where $\alpha > 0$ is a system parameter to ensure that the utility is greater than or equal to zero under the optimal R^*.

In a large-scale case, it is difficult for a single drone to cover the entire area. Therefore, a dynamic incentive mechanism can be designed to select the optimal clients in adaptation to the time-varying environment. The difference from the static incentive is that C in the dynamic incentive represents the computing cost of the client to complete a round of local training. In the dynamic scene, we use r^{τ} instead of R in the formula, where r^{τ} represents the reward determined by the drone before the τth global model is updated. For convenience, in the following analysis, we uniformly use R to express the reward.

The decision making problem can be modelled using the Stackelberg game. In the game, the DT of the drone acts as the leader, while the ground clients are the follower. The game consists of two stages. In the first stage, the aggregator publishes the task and determines its reward R. In the second stage, each client will devise strategies to determine the number of rounds to participate in federated learning and maximize their respective utilities [105]. The second stage of the Stackelberg game is a noncooperative game, that is, in which there is a Nash equilibrium. A set of strategies $\mathcal{T}^* = (\tau_1^*, \tau_2^*, \cdots, \tau_N^*)$ is a Nash equilibrium in the second stage of the game if, for any client i, $U_i(\tau_i^*, \tau_{-i}^*) \geq U_i(\tau_i, \tau_{-i}^*)$, $\forall \tau_i > 0$. Under the reward R given by the aggregator, no client can gain any additional benefits by unilaterally changing the current strategy.

According to Nash equilibrium, when all the other clients expect client i to play their best strategy, client i can only play τ_i^*. Therefore, we need to introduce the concept of the best response strategy. Given τ_{-i}, a strategy is client i's best response strategy, denoted by $\beta_i(\tau_{-i})$, if it maximizes $U_i(\tau_i, \tau_{-i})$ over all $\tau_i \geq 0$. To find the Nash equilibrium in the second stage of the game, a closed-form solution of the best response strategy for each client must be calculated. Accordingly, if the whole game has a unique Stackelberg equilibrium, the necessary and sufficient condition is for there to be a unique optimal solution in the first stage of the game. There exists a unique Stackelberg equilibrium $(R^*, \mathcal{T}^*)$, where R^* is the only value that can maximize the utility of the aggregator over $R \in [0, \infty)$. The utility function of the aggregator is a concave quadratic function on the difference between the reward and loss of client i, and the first derivative of the utility function is equal to zero. Then the optimal R can be solved. At this time, $(R^*, \mathcal{T}^*)$ is the unique Stackelberg equilibrium in the game.

Different from the static mechanism, the dynamic mechanism selects clients according to the ratio of the unit local training computing cost and reputation value, that is, $\frac{c_i}{\rho_i}$. The drone's optimal payment R^* should be expressed as $R^* = \sum_{\tau=1}^{\tau_g} (r^\tau)^*$, where τ_g is the number of rounds of the global update. Finally, t_i^* and $(r^\tau)^*$ constitute the unique equilibrium of the Stackelberg game.

7.4.4 Illustration of the Results

We use the software Pytorch 0.4.1 to build a federated learning model in an air–ground network and use the classic Modified National Institute of Standards and Technology data set to evaluate the performance of the proposed incentive mechanisms. We set up a total of 10 to 100 clients. Under the dynamic incentive, the communications range of a drone can only cover 20 clients at the same time. We employ a cost-only scheme as the benchmark where clients with low training costs are selected to participate in federated learning.

Figure 7.6 shows the model's accuracy with varying global update rounds under three schemes. The convergence accuracy of the global model relies on the participating clients and their data quality. The accuracy under the dynamic incentive scheme is the highest. After each round of global updates, the performance of the clients will be evaluated, and the participation of low-quality clients will be reduced.

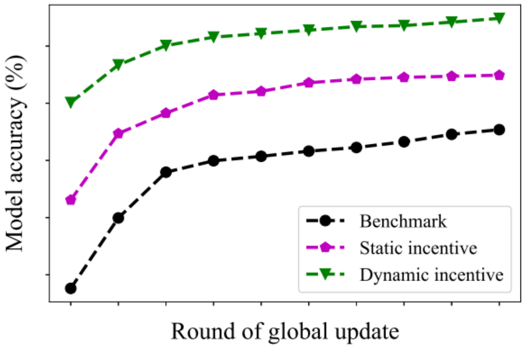

Fig. 7.6 Comparison of model accuracy under varying global update rounds

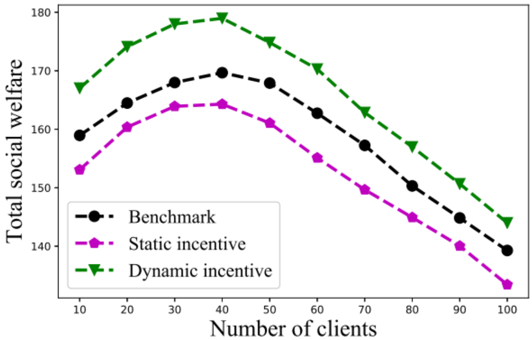

Fig. 7.7 The total social welfare of a drone and clients varies with the number of clients

The static scheme chooses the optimal client set, which might not be appropriate later in the federated learning process due to the mobility of the drone. Thus, the accuracy of the static incentive is lower than that of the dynamic incentive. Since the benchmark considers only the training costs of the clients, its model accuracy is 5% lower than that of the static scheme.

Figure 7.7 compares the total social welfare of the drone and clients varies with the number of clients under three schemes. As the total number of clients increases, the total social welfare increases first, peaks at around 40 clients, and then decreases. With the increase of the client number, the utility of the drone increases, while the utilities of the clients decrease due to the greater number of competitors. In addition, the benchmark social welfare is higher than that of the static incentive, because the benchmark selects only clients with low cost. Thus, its social welfare is the highest among the three schemes.