Multi-Functional Time Expanded Graph: A Unified Graph Model for Communication, Storage, and Computation for Dynamic Networks Over Time

Wei Liu, *Senior Member, IEEE*, Huiting Yang, and Jiandong Li, *Fellow, IEEE*

***Abstract*— Space-air-ground integrated network (SAGIN) aided multi-tier computing network can be modelled as a dynamic and predictable network. For the SAGIN aided multi-tier computing network, the traditional time expanded graph (TEG) can only jointly model communication and storage capability, as well as one computing function for one mission flow within one same node. However, for multiple computing functions for one mission flow in one same node, TEG is not applicable. In this paper, for SAGIN aided multi-tier computing networks, we propose an multi-functional time expanded graph (MF-TEG) to jointly model the communication, storage, and computation capability of nodes where multiple computing functions for one mission flow in one same node can be characterized. Specifically, based on TEG, for each node having computation functions, we adopt the virtual network graph (VNG) to virtually decompose it into three virtual components: sub-virtual node, virtual computing nodes, and virtual transmission links, where the virtual computing node provides the computing function. We characterize the amount of data flow on each link and also present four kinds of fundamental constraints for the data flow in the MF-TEG for joint communication, storage, and computing function: computation capacity constraints, communication capacity constraints, storage capacity constraints, and flow conservation constraints. We provide one example of using MF-TEG to model the SAGIN aided multi-tier computing network with a service function chain (SFC), where satellite nodes could provide communication, storage, and multiple computing functions for one mission flow in one same node, where TEG is not valid. Furthermore, simulation results show that for SAGIN aided multi-tier computing network, the proposed MF-TEG model significantly outperforms the snapshot graph-aided VNG (SSG-aided VNG) model. The reason for that is only communication and computation capability is considered by the SSG-aided VNG model, while storage capability is not exploited.**

***Index Terms*— Dynamic network, time expanded graph, virtual network graph, multi-functional time expanded graph, communication, storage, computation.**

I. Introduction

THE Internet of Thing (IoT) plays a critical role in future 6G [1], [2]. In order to support IoT service, multi-tier computing is required [3], [4], [5], [6], [7]. Specifically, multi-tier computing integrates cloud, fog, and edge computing technologies, which can provide flexible computing capability for IoT [3], [4], [5]. Furthermore, one of the key features in 6G is the space-air-ground integrated network (SAGIN) [8], [9], [10]. The SAGIN is a multidimensional heterogeneous network consisting of three network segments, i.e., the space network (SN) composed of satellites, the aerial network composed of high altitude platforms (HAPs) and flying unmanned aerial vehicles (UAVs), and the terrestrial network composed of devices and ground stations [11], [12]. One of the important application scenarios of the SAGIN is to provide multi-tier computing services for IoT devices in remote areas where there is no cellular coverage [2], [13], [14], [15], [16]. Specifically, in the aerial network, the flying UAVs can serve as edge servers to provide edge computing capabilities for ground users [2], [13], while in the SN, satellites can provide access to cloud computing service through seamless network coverage [2], [13]. Hence, the IoT devices in remote areas can offload their missions with computation requirements, such as monitoring and video surveillance, to their nearby UAVs for edge computing or satellites for access to the cloud computing [2], [13], [14], [15].

In SAGIN aided multi-tier computing, the UAVs can be configured with fixed flying trajectories to serve the considered area [13], [17]. Moreover, satellites move dynamically and their moving orbits are determined [18], [19], which results in the intermittent yet predictable connection among satellites in SAGIN [13], [17]. In this case, the SAGIN aided multi-tier computing network consisting of UAVs and satellites can be characterized by time-varying and predictable network topology.

Recently, time-varying graphs, such as the snapshot sequence graph (SSG) and time expanded graph (TEG), have been proposed to describe the time-varying and predictable network topology [17], [18], [20]. Specifically, the SSG can be used to describe the dynamic evolution of the topology through the snapshots in discrete timeslots [21]. However, SSG ignores the connection among snapshots, i.e., without considering the storage capacity, resulting in low resource utilization [19]. The TEG, proposed by Ford and Fulkerson [22], [23], connects the snapshots in discrete timeslots by introducing storage links.

Joint communication, computing, and storage modelling has been widely investigated for both static networks [24], [25], [26] and dynamic networks [27], [28], [29], [30], [31]. In [25], joint service function placement and flow distribution

Manuscript received 11 May 2022; revised 2 September 2022; accepted 25 October 2022. Date of publication 5 January 2023; date of current version 19 January 2023. This work was supported in part by the China National Key Research and Development Program under Grant YFA1000500, in part by the National Natural Science Foundation of China under Grant 61871452, and in part by the Key Research and Development Program of Shaanxi under Grant 2022ZDLGY05-05. *(Corresponding author: Huiting Yang.)*

The authors are with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, Shaanxi 710071, China (e-mail: liuweixd@mail.xidian.edu.cn; htyang0827@stu.xidian.edu.cn; jdli@mail.xidian.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2022.3233533.

Digital Object Identifier 10.1109/JSAC.2022.3233533

TABLE I

OUR CONTRIBUTIONS AGAINST THE STATE-OF-ART MODELS

	Our Proposed MF-TEG	Traditional TEG [22]	VNG [24]
Communication capability	√	√	√
Storage capability	√	√	
Computing capability	√	√	√
Characterize multiple computing functions for one mission flow within one same node	√		√
Support optimizing dynamic network over time	√	√	

with service function chain (SFC) constraints was investigated, which jointly considers the communication and computing capability of nodes, without considering the storage capability. In [26], joint communication and computing capability of nodes are considered in the service-oriented network with SFC constraints, without considering the storage capability. However, in the above works [25], [26], the computation capability can only be modelled for the case where each mission flow receives at most one computing function in one same node. In fact, one mission flow may receive multiple computing functions in one same node [24]. In this case, the traditional static graph cannot characterize the flow conservation constraints of mission flow receiving multiple computing functions in one same node [24]. To solve this problem, a virtual network graph (VNG) was proposed to characterize the flow conservation constraints of mission flow receiving multiple computing functions in one same node [24]. However, in [24], by introducing VNG, only the joint communication and computing capability of nodes are modelled in static networks without considering the storage capability.

For dynamic networks, in [27], the SSG is used to model the time-varying topology of space network, which jointly model the communication and computing capability of satellites without considering the storage capability. In [28], the TEG is used to model the dynamic topology of the device-to-device network, where joint communication and storage capability of nodes are considered without considering the computing capability. In [32], the TEG is used to model the dynamic topology of the delay-tolerant aeronautical ad hoc network (AANet), where joint communication and storage capability of nodes are considered without considering the computing capability. In [29], [30], and [31], the TEG is used to model the time-varying topology of the space network, which jointly models the communication, storage, and computing capability of satellites [29], [30], [31]. However, in [29], [30], and [31], they consider the scenario where each mission flow receives at most one function in one same node. In fact, one drawback of the traditional TEG is that it cannot characterize the flow conservation constraints of mission flow receiving multiple computing functions in one same node [24].

Against this background, in this treatise, we propose one powerful framework, the multi-functional time expended graph (MF-TEG), which is capable of modelling the communication, storage, and computing capability in the dynamic network over time, where multiple computing functions for one mission flow in one same node can be characterized, instead of only communication and storage as well as only one computing fucntion for one mission flow in one same node in the traditional TEG, or the only communication and computing capability in VNG [24]. In Table I, we boldly contrast our contributions to the state-of-the-art. Our main contributions are as follows:

- For the dynamic network, we propose the framework of MF-TEG to jointly model communication, storage, and computation capability. Specifically, based on TEG, for each node having computing functions, we adopt VNG [24] to virtually decompose it into three virtual components: sub-virtual node, virtual computing nodes, and virtual transmission links, where the sub-virtual node keeps the communication and storage capability of the original node, while virtual computing node provides the computing capability of the original node, and the virtual transmission links connect the sub-virtual node and virtual computing node.
- We characterize the amount of data flow on each link and also present four kinds of fundamental constraints for the data flow in the MF-TEG for joint communication, storage, and computing function: computation capacity constraints, communication capacity constraints, storage capacity constraints and flow conservation constraints. Especially, for multiple computing functions for one mission flow in one same node, by using the proposed MF-TEG model, flow conservation constraints can be constructed for joint communication, storage, and computing capability.
- The proposed MF-TEG can find many applications. We provide one example for the scenario of SAGIN aided multi-tier computing network. Specifically, the device offloads its mission flow to the cloud computing server through the SN. This mission flow invokes the SFC which consists of a predefined sequence of virtual network functions (VNFs). Each VNF accommodates one specific function, which requires computing capability. In this case, in SN with SFC, satellite nodes deployed with VNFs provide communication, storage, and computing function. Furthermore, multiple VNFs for one mission flow, i.e., multiple computing functions for one mission flow in one same node, could be needed within one same node. Therefore, the traditional TEG is not applicable. Furthermore, for the maximum flow, the simulation results show that the performance of the proposed MF-TEG is much higher than that provided by VNG, as VNG does not exploit the storage capability.

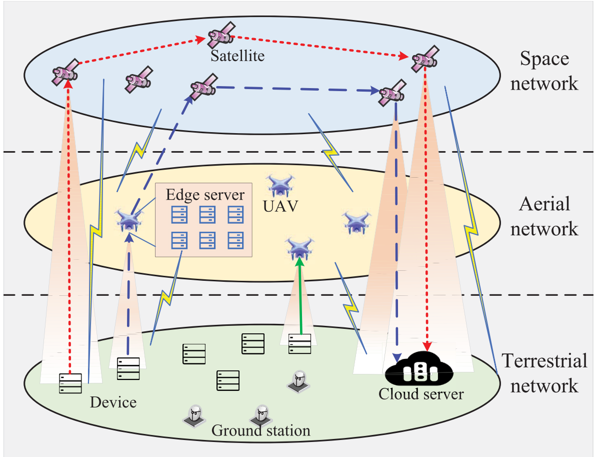


Fig. 1. The SAGIN aided multi-tier computing network.

II. System Model

A. Dynamic and Predictable Network Model

The SAGIN aided multi-tier computing network consists of UAVs and satellites as shown in Fig. 1, where the computing tasks of devices can be offloaded to UAV edge computing service, as shown as green solid lines in Fig. 1, or can be offloaded to the cloud computing server through satellites as shown as red dotted lines in Fig. 1, or can be offloaded to the cloud server through UAVs and satellites, where the UAV acts as a relay node, as shown as blue dashed lines in Fig. 1. As UAVs can be configured with fixed flying trajectories [13], [17], and moving orbits of satellites are determined [18], [19], the SAGIN aided multi-tier computing network can be modelled as a dynamic and predictable network [13], [17].

Assume that one dynamic and predictable network consists of N nodes, which could be the UAVs or satellites, as shown in Fig. 2. For the given time horizon $\mathcal{T} = [t_0, t_T)$, according to the connectivity between nodes, it can be divided into T consecutive timeslots $\{\tau_1, \tau_2, \cdots, \tau_q, \cdots, \tau_T\}$, where $\tau_q = [t_{q-1}, t_q)(\forall q, 1 \leq q \leq T)$ and the network topology is fixed during timeslot τ_q, but may be different in different timeslots [22], [23].

For different timeslot, the nodes may have different connection status, as shown in Fig. 2. For each timeslot, the dynamic network can be modelled by a snapshot graph $\mathcal{G}_\mathcal{S} = (\mathcal{V}_\mathcal{S}, \mathcal{L}_\mathcal{S})$, where $\mathcal{V}_\mathcal{S}$ denotes the set of nodes in the network, while $\mathcal{L}_\mathcal{S}$ denotes the set of links in the network for the current timeslot, i.e.,

$$\begin{aligned} \mathcal{V}_\mathcal{S} &= \{v_i \mid 1 \leq i \leq N\}, \quad (1)\\ \mathcal{L}_\mathcal{S} &= \big\{ (v_i, v_j) \mid v_i \text{ can transmit data to } v_j \text{ at the} \\ &\qquad \text{current timeslot } \tau_q \big\}, \quad (2) \end{aligned}$$

where v_i represents the ith node in the network and (v_i, v_j) denotes the transmission link from node v_i to node v_j, as shown as blue solid lines in Fig. 2.

However, based on the snapshot, the network performance can only be considered for each timeslot, while ignoring the connection among different timeslots.

B. Traditional Time Expended Graph

By adding the storage link between two consecutive timeslots, as shown in dotted red lines in Fig. 3, the dynamic network among different timeslots can be connected over time. This kind of connection of dynamic network over time can be modelled as the TEG [22], [23] composed of T layers, where each layer corresponds to a network topology of one timeslot [22], [23], as shown in Fig. 3. Denote the TEG as $\mathcal{G}_\mathcal{T} = (\mathcal{V}_\mathcal{T}, \mathcal{L}_\mathcal{T})$, where $\mathcal{V}_\mathcal{T}$ and $\mathcal{L}_\mathcal{T}$ represent the set of nodes and links in the TEG, respectively.

1) Vertices: The nodes of the TEG correspond to the replicas of nodes in the dynamic network for each time slot. Specifically, at the timeslot τ_q, node v_i in the dynamic network is represented by node $v_i^{\tau_q}$ in the TEG. Therefore, the set of nodes in the TEG is denoted by

$$\mathcal{V}_\mathcal{T} = \left\{ v_i^{\tau_q} \middle| v_i \in \mathcal{V}, 1 \leq q \leq T, 1 \leq i \leq N \right\}. \quad (3)$$

2) Links: There are two different types of directed links in the TEG, namely *transmission links* and *storage links* [20], [22], [23]. The set of *transmission links* in the TEG is denoted as $\mathcal{L}_{\mathcal{T},t}$, as shown as blue solid lines in Fig. 3, which is given by

$$\mathcal{L}_{\mathcal{T},t} = \Big\{ \left(v_i^{\tau_q}, v_j^{\tau_q}\right) \mid 1 \leq q \leq T, v_i \text{ can transmit data to } v_j \text{ during timeslot } \tau_q \Big\}. \quad (4)$$

The set of *storage links* in the TEG is denoted as $\mathcal{L}_{\mathcal{T},s}$, as shown as red dashed lines between the consecutive timeslots in Fig. 3, which is given by

$$\mathcal{L}_{\mathcal{T},s} = \left\{ \left(v_i^{\tau_q}, v_i^{\tau_{q+1}}\right) \mid v_i \in \mathcal{V}, 1 \leq q \leq T-1 \right\}. \quad (5)$$

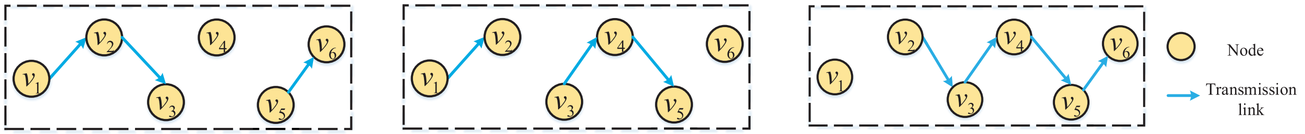
Specifically, *storage link* $\left(v_i^{\tau_q}, v_i^{\tau_{q+1}}\right)$ represents that node v_i in the dynamic network can store its data from timeslot τ_q to timeslot τ_{q+1}. Consequently, $\mathcal{L}_\mathcal{T}$ can be represented as $\mathcal{L}_\mathcal{T} = \mathcal{L}_{\mathcal{T},t} \cup \mathcal{L}_{\mathcal{T},s}$.

Denote the communication capacity of a *transmission link* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$ as $C_{\mathcal{T},t}\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$, which represents the maximum amount of data that can be transmitted by the link within the timeslot τ_q. Denote the storage capacity of a *storage link* $\left(v_i^{\tau_q}, v_i^{\tau_{q+1}}\right)$ as $C_{\mathcal{T},s}\left(v_i^{\tau_q}, v_i^{\tau_{q+1}}\right)$, which represents the maximum amount of data which can be stored by node v_i in the dynamic network from timeslot τ_q to timeslot τ_{q+1}.

Based on the TEG, the network performance can be optimized over the time horizon, which is capable of improving the system performance against the snapshot model.

III. Multi-Functional Time Expanded Graph

In TEG, both communication and storage capabilities can be modelled. However, some nodes in the network also have computing capability, for example, one image is received in one node and then compressed, and transmitted to another node. Therefore, the nodes in the network can be divided into two types: computing function nodes which can provide processing functions, and non-computing function nodes which cannot provide processing functions. Denote $\mathcal{V}_c$ and $\mathcal{V}_{nc}$ as the set of



(a) The first snapshot (b) The second snapshot (c) The third snapshot

Fig. 2. An example of the snapshots of dynamic network with $N = 6$ nodes for three timeslots.

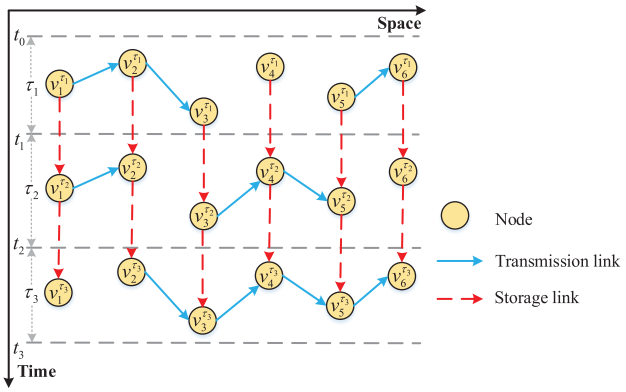


Fig. 3. Schematic of Time Expanded Graph.

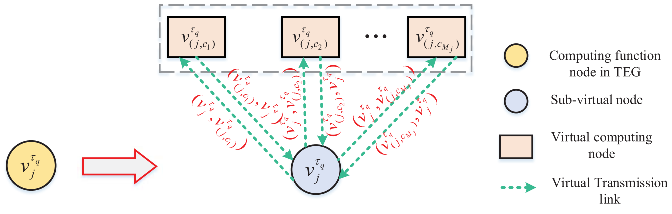

$v_j^{\tau_q}$
$v_{(j,c_1)}^{\tau_q}$
$v_{(j,c_2)}^{\tau_q}$
$v_{(j,c_{M_j})}^{\tau_q}$
$(v_j^{\tau_q}, v_{(j,c_1)}^{\tau_q})$
$(v_{(j,c_1)}^{\tau_q}, v_j^{\tau_q})$
$(v_j^{\tau_q}, v_{(j,c_2)}^{\tau_q})$
$(v_{(j,c_2)}^{\tau_q}, v_j^{\tau_q})$
$(v_j^{\tau_q}, v_{(j,c_{M_j})}^{\tau_q})$
$(v_{(j,c_{M_j})}^{\tau_q}, v_j^{\tau_q})$
Computing function node in TEG
Sub-virtual node
Virtual computing node
Virtual Transmission link

Fig. 4. Schematic of Decoupling a node into three types of components: sub-virtual node, virtual computing node and virtual transmission links.

computing function nodes and non-computing function nodes in the network, respectively, which are given, respectively, as

$$\mathcal{V}_c = \{v_i \mid \text{node } v_i \text{ has some computing functions}\}, \tag{6}$$

$$\mathcal{V}_{nc} = \{v_i \mid \text{node } v_i \text{ cannot provide any computing function}\}. \tag{7}$$

The TEG can only model some simple computing capability of nodes. For example, in [20], [31], for one mission flow, only one computing function within one same node is modelled. However, multiple computing functions for one mission flow in one same node may be needed. In this case, TEG is not valid for modelling computing capability. Hence, we propose the MF-TEG to jointly model the communication, storage, and computation capability, where multiple computing functions for one mission flow in one same node may be needed.

A. Multiple Commodities and Multiple Functions

Assume that at time slot τ_q, there are $N_j^{\tau_q}$ kinds of data flows (i.e., $N_j^{\tau_q}$ kinds of commodities) going into node $v_j^{\tau_q}$ through all the *transmission links* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{T},t}$ and the *storage link* $\left(v_j^{\tau_q - 1}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{T},s}$. Denote $\mathcal{A}_j^{\tau_q}$ as the set of different kinds of data flows going into node $v_j^{\tau_q}$ through all the *transmission links* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{T},t}$ and the *storage link* $\left(v_j^{\tau_q - 1}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{T},s}$ going into node $v_j^{\tau_q}$ at time slot τ_q, which is given by

$$\begin{aligned} \mathcal{A}_j^{\tau_q} = \Big\{ & \xi_n \mid \text{the data flow } \xi_n \text{ going into the node } v_j^{\tau_q} \text{ through} \\ & \text{all the } \textit{transmission links } \left(v_i^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{T},t}, \text{ or} \\ & \text{the } \textit{storage link } \left(v_j^{\tau_q - 1}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{T},s}, n \in [N_j^{\tau_q}] \Big\}, \end{aligned} \tag{8}$$

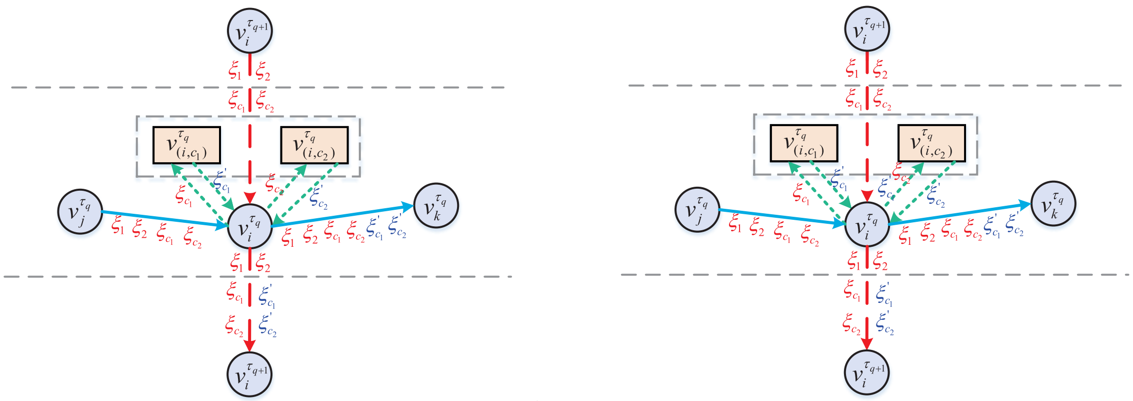
where, ξ_n ($\forall n \in [N_j^{\tau_q}]$) represents the nth kind of data flow. Hence, $N_j^{\tau_q} = |\mathcal{A}_j^{\tau_q}|$. Furthermore, node $v_j^{\tau_q}$ can provide computing functions for M_j out of $N_j^{\tau_q}$ kinds of data flows $\{\xi_{c_m}\}_{m \in [M_j], c_m \in [N_j^{\tau_q}]}$.

Furthermore, denote the computing function needed by data flow ξ_{c_m} as $f_{\xi_{c_m}, \xi'_{c_m}}$, where ξ'_{c_m} denotes the output data flow by the computing function $f_{\xi_{c_m}, \xi'_{c_m}}$ for input data flow ξ_{c_m}, i.e., the data flow ξ_{c_m} is transferred into data flow ξ'_{c_m} once it goes through computing function $f_{\xi_{c_m}, \xi'_{c_m}}$.

In order to model the computing capability of node $v_j^{\tau_q}$, VNG is adopted, where multiple computing functions for one mission flow in one same node can be characterized [24]. Specifically, by introducing VNG, a node $v_j^{\tau_q}$ is virtually decoupled into several sub-components, that is, sub-virtual node $v_j^{\tau_q}$ and the M_j virtual computing nodes $\{v_{(j,c_m)}^{\tau_q}\}_{m \in [M_j]}$, as well as virtual transmission links $\{(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q})\}_{m \in [M_j]}$ and $\{(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q})\}_{m \in [M_j]}$ [24], as shown in Fig.4. Specifically, the sub-virtual node $v_j^{\tau_q}$ takes the communication function and storage functions of node $v_j^{\tau_q}$, and $v_{(j,c_m)}^{\tau_q}$ denotes the virtual computing node which provides the computing function $f_{\xi_{c_m}, \xi'_{c_m}}$ of node $v_j^{\tau_q}$, while $(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q})$ and $(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q})$ denote the virtual transmission links from the sub-virtual node $v_j^{\tau_q}$ to the virtual computing node $v_{(j,c_m)}^{\tau_q}$ and from the virtual computing node $v_{(j,c_m)}^{\tau_q}$ to the sub-virtual node $v_j^{\tau_q}$, respectively.

Furthermore, the sub-virtual node $v_j^{\tau_q}$ forwards some of the received data to node $v_j^{\tau_q}$'s neighbor nodes and stores the rest of them in the storage of node $v_j^{\tau_q}$. The virtual computing node $v_{(j,pu_m)}^{\tau_q}$ takes some of the received data flow ξ_{c_m} from sub-virtual node $v_j^{\tau_q}$, and then processes this data flow (for example, image compression) and transfers it into date flow ξ'_{c_m}. Its output is the processed data flow by using the computing function $f_{\xi_{c_m}, \xi'_{c_m}}$.

Moreover, this decomposition can characterize both multiple parallel computing functions and multiple successive computing functions in a node. For multiple parallel computing functions, each virtual computing node only takes the data flow from the sub-virtual node, not from other virtual computing nodes, as shown in Fig. 5(a), where ξ'_{c_1} is different from



(a) Two parallel computing functions in a node, where data flow ξ_{c_1}' is different from data flow ξ_{c_2}

(b) Two successive computing functions in a node,where data flow ξ_{c_1}' is the same as data flow ξ_{c_2}

Fig. 5. Examples of Multiple Parallel and Successive Computing Functions.

ξ_{c_2}, and the virtual computing function node $v_{(j,c_2)}^{\tau_q}$ can only take the output data flow ξ_{c_2} from sub-virtual node $v_j^{\tau_q}$, not the ξ_{c_1}' from virtual computing function node $v_{(j,c_1)}^{\tau_q}$, as its input data flow. However, for multiple successive computing functions, one virtual computing node may take the data flow both from the sub-virtual node and other virtual computing nodes, as shown in Fig. 5(b), where ξ_{c_1}' is the same as ξ_{c_2} and the virtual computing function node $v_{(j,c_2)}^{\tau_q}$ takes the output data flow ξ_{c_1}' from both virtual computing function node $v_{(j,c_1)}^{\tau_q}$ and ξ_{c_2} from sub-virtual node $v_j^{\tau_q}$ as its input data flow and transform them into data flow ξ_{c_2}'.

Remark 1: By characterizing two successive computing functions in one node, the MF-TEG is capable of modelling the computing capability of multiple computing functions for one mission flow within one same node.

Therefore, the TEG in Fig. 3 is transferred into MF-TEG as shown in Fig. 6. Denote the MF-TEG as $\mathcal{G}_{\mathcal{F}} = (\mathcal{V}_{\mathcal{F}}, \mathcal{L}_{\mathcal{F}})$, where $\mathcal{V}_{\mathcal{F}}$ and $\mathcal{L}_{\mathcal{F}}$ represent the set of nodes and links in the MF-TEG, respectively.

1) Vertices: There are three kinds of vertices in MF-TEG, i.e., non-computing function nodes, sub-virtual nodes, and virtual computing nodes. Therefore, the set of nodes in the MF-TEG also can be expressed as

$$\mathcal{V}_{\mathcal{F}} = \mathcal{V}_{\mathcal{F},nc} \cup \mathcal{V}_{\mathcal{F},sv} \cup \mathcal{V}_{\mathcal{F},c}, \tag{9}$$

where $\mathcal{V}_{\mathcal{F},nc}$ denotes the set of non-computing function nodes, $\mathcal{V}_{\mathcal{F},sv}$ denotes the set of sub-virtual nodes, while $\mathcal{V}_{\mathcal{F},c}$ denotes the set of virtual computing nodes, respectively.

a) **Non-computing function node**: it cannot provide computing functions in the MF-TEG, which corresponds to the replica of the non-computing function node in TEG. The set of the non-computing function nodes $\mathcal{V}_{\mathcal{F},nc}$ in the MF-TEG can be expressed as

$$\mathcal{V}_{\mathcal{F},nc} = \left\{ v_j^{\tau_q} \mid v_j \in \mathcal{V}_{nc}, 1 \leq q \leq T \right\}. \tag{10}$$

b) **Sub-virtual node**: it only maintains the communication function and storage functions of computing nodes and cannot provide computing functions in the MF-TEG, which corresponds to the replica of the computing function node in TEG. The set of the sub-virtual nodes $\mathcal{V}_{\mathcal{F},sv}$ in the MF-TEG can be expressed as

$$\mathcal{V}_{\mathcal{F},sv} = \left\{ v_j^{\tau_q} \mid v_j \in \mathcal{V}_c, 1 \leq q \leq T \right\}. \tag{11}$$

c) **Virtual computing node**: it provides the computing function in the MF-TEG. The set of the virtual computing nodes $\mathcal{V}_{\mathcal{F},c}$ in the MF-TEG can be expressed as

$$\mathcal{V}_{\mathcal{F},c} = \left\{ v_{(j,c_m)}^{\tau_q} \mid v_j \in \mathcal{V}_c, 1 \leq m \leq M_j, 1 \leq q \leq T \right\}. \tag{12}$$

2) Links: There are three different types of directed links of the MF-TEG, i.e., *transmission links*, *storage links*, and *virtual transmission links*. Therefore, the set of links $\mathcal{L}_{\mathcal{F}}$ in the MF-TEG can be expressed as

$$\mathcal{L}_{\mathcal{F}} = \mathcal{L}_{\mathcal{F},t} \cup \mathcal{L}_{\mathcal{F},s} \cup \mathcal{L}_{\mathcal{F},c}, \tag{13}$$

where $\mathcal{L}_{\mathcal{F},t}$ denotes the set of *transmission links*, $\mathcal{L}_{\mathcal{F},s}$ denotes the set of *storage links*, and $\mathcal{L}_{\mathcal{F},c}$ denotes the set of *virtual transmission links*, respectively.

a) ***Transmission link***: it models the communication capability between nodes in MF-TEG, as shown as blue solid lines in Fig. 6. The set of *transmission links* $\mathcal{L}_{\mathcal{F},t}$ in the MF-TEG, can be expressed as

$$\mathcal{L}_{\mathcal{F},t} = \Big\{ \left(v_i^{\tau_q}, v_j^{\tau_q}\right) \mid 1 \leq q \leq T, v_i \text{ can transmit data flows to } v_j \text{ during timeslot } \tau_q \Big\}. \tag{14}$$

Furthermore, denote $C_{\mathcal{F},t}\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$ as the communication capacity of a *transmission link* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$ in the MF-TEG, which represents the maximum amount of data that can be transmitted by the *transmission link* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$ during the timeslot τ_q [20], [33]. Moreover, as the communication function of the node $v_j^{\tau_q}$ in TEG is kept unchanged by sub-virtual node $v_j^{\tau_q}$ in MF-TEG, we have

$$\mathcal{L}_{\mathcal{F},t} = \mathcal{L}_{\mathcal{T},t}, \tag{15}$$

$$C_{\mathcal{F},t}\left(v_i^{\tau_q}, v_j^{\tau_q}\right) = C_{\mathcal{T},t}\left(v_i^{\tau_q}, v_j^{\tau_q}\right), \ \forall \left(v_i^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},t}. \tag{16}$$

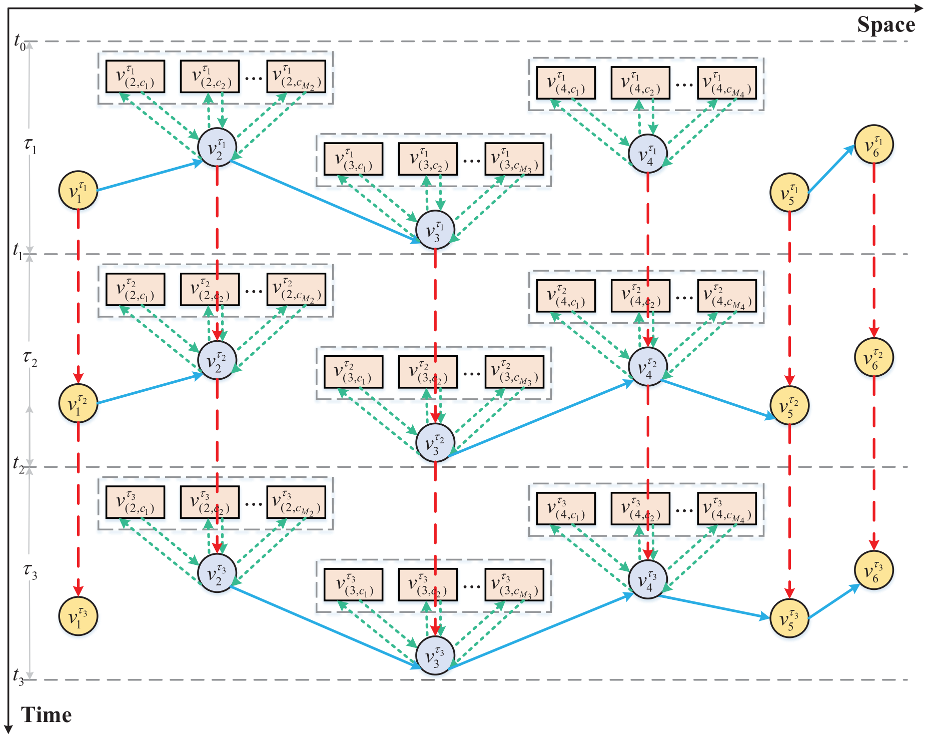

Space
Time
t_0
t_1
t_2
t_3
τ_1
τ_2
τ_3
$v_1^{\tau_1}$
$v_2^{\tau_1}$
$v_3^{\tau_1}$
$v_4^{\tau_1}$
$v_5^{\tau_1}$
$v_6^{\tau_1}$
$v_{(2,c_1)}^{\tau_1}$ $v_{(2,c_2)}^{\tau_1}$ ··· $v_{(2,c_{M_2})}^{\tau_1}$
$v_{(3,c_1)}^{\tau_1}$ $v_{(3,c_2)}^{\tau_1}$ ··· $v_{(3,c_{M_3})}^{\tau_1}$
$v_{(4,c_1)}^{\tau_1}$ $v_{(4,c_2)}^{\tau_1}$ ··· $v_{(4,c_{M_4})}^{\tau_1}$
$v_1^{\tau_2}$
$v_2^{\tau_2}$
$v_3^{\tau_2}$
$v_4^{\tau_2}$
$v_5^{\tau_2}$
$v_6^{\tau_2}$
$v_{(2,c_1)}^{\tau_2}$ $v_{(2,c_2)}^{\tau_2}$ ··· $v_{(2,c_{M_2})}^{\tau_2}$
$v_{(3,c_1)}^{\tau_2}$ $v_{(3,c_2)}^{\tau_2}$ ··· $v_{(3,c_{M_3})}^{\tau_2}$
$v_{(4,c_1)}^{\tau_2}$ $v_{(4,c_2)}^{\tau_2}$ ··· $v_{(4,c_{M_4})}^{\tau_2}$
$v_1^{\tau_3}$
$v_2^{\tau_3}$
$v_3^{\tau_3}$
$v_4^{\tau_3}$
$v_5^{\tau_3}$
$v_6^{\tau_3}$
$v_{(2,c_1)}^{\tau_3}$ $v_{(2,c_2)}^{\tau_3}$ ··· $v_{(2,c_{M_2})}^{\tau_3}$
$v_{(3,c_1)}^{\tau_3}$ $v_{(3,c_2)}^{\tau_3}$ ··· $v_{(3,c_{M_3})}^{\tau_3}$
$v_{(4,c_1)}^{\tau_3}$ $v_{(4,c_2)}^{\tau_3}$ ··· $v_{(4,c_{M_4})}^{\tau_3}$

Fig. 6. Schematic of Multi-functional Time Expanded Graph.

b) ***Storage link***: it models the storage capability of nodes in MF-TEG, as shown as red dashed lines in Fig. 6. The set of *storage links* $\mathcal{L}_{\mathcal{F},s}$ in the MF-TEG can be expressed as

$$\mathcal{L}_{\mathcal{F},s} = \left\{ \left(v_j^{\tau_q}, v_j^{\tau_{q+1}} \right) \mid v_j \in \mathcal{V}, 1 \le q \le T-1 \right\}. \quad (17)$$

Furthermore, denote $\mathcal{C}_{\mathcal{F},s}\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right)$ as the storage capacity of a *storage link* $\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right)$ in the MF-TEG, which represents the maximum amount of data that can be stored by node $v_i^{\tau_q}$ [20], [33], [34]. Moreover, as the storage function of the node $v_j^{\tau_q}$ in TEG is kept unchanged by sub-virtual node $v_j^{\tau_q}$ in MF-TEG, we have

$$\mathcal{L}_{\mathcal{F},s} = \mathcal{L}_{\mathcal{T},s}, \quad (18)$$

$$C_{\mathcal{F},s}\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right) = C_{\mathcal{T},s}\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right), \quad \forall \left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right) \in \mathcal{L}_{\mathcal{F},s}. \quad (19)$$

c) ***Virtual transmission link***: it models the virtual connection between the sub-virtual node and the virtual computing node in MF-TEG, as shown as green dotted lines in Fig. 6. Furthermore, the set of *virtual transmission links* in the MF-TEG can be expressed as

$$\mathcal{L}_{\mathcal{F},c} = \mathcal{L}_{\mathcal{F},c}^{in} \cup \mathcal{L}_{\mathcal{F},c}^{out}, \quad (20)$$

where $\mathcal{L}_{\mathcal{F},c}^{in}$ denotes the *virtual transmission link* from the sub-virtual node to the virtual computing node, while $\mathcal{L}_{\mathcal{F},c}^{out}$ denotes the *virtual transmission link* from the virtual computing node to the sub-virtual node, which are given by

$$\mathcal{L}_{\mathcal{F},c}^{in} = \left\{ \left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q} \right) \mid v_j \in \mathcal{V}_c, 1 \le m \le M_j, 1 \le q \le T \right\}, \quad (21)$$

$$\mathcal{L}_{\mathcal{F},c}^{out} = \left\{ \left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q} \right) \mid v_j \in \mathcal{V}_c, 1 \le m \le M_j, 1 \le q \le T \right\}. \quad (22)$$

Furthermore, denote $C_{\mathcal{F},\xi_n}\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right)$ and $C_{\mathcal{F},\xi_n}\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right)$ as the communication capacity of *virtual transmission links* $\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right)$ and $\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right)$ for the data flow ξ_n in the MF-TEG, respectively, which represent the maximum amount of data flow ξ_n that can be transmitted through the corresponding *virtual transmission links* $\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right)$ and $\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right)$ during the timeslot τ_q, respectively.

Moreover, as the virtual computing node $v_{(j,c_m)}^{\tau_q}$ only provides the computing function $f_{\xi_{c_m},\xi'_{c_m}}$, only data flow ξ_{c_m} is allowed to use the *virtual transmission link* $\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right)$, while only data flow ξ'_{c_m} is allowed to use the *virtual transmission link* $\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right)$. Consequently, the communication capacity $C_{\mathcal{F},\xi_n}\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right)$ and $C_{\mathcal{F},\xi_n}\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right)$ are given by

$$C_{\mathcal{F},\xi_n}\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right) = \begin{cases} \infty, & n = c_m, \\ 0, & n \ne c_m, \end{cases} \forall \left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},c}^{in}, \quad (23)$$

$$C_{\mathcal{F},\xi'_n}\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right) = \begin{cases} \infty, & n = c_m, \\ 0, & n \ne c_m, \end{cases} \forall \left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},c}^{out}. \quad (24)$$

B. The Amount of Data Flow on Each Link

1) For Transmission Link: Assuming there are $N_{i,j}^{\tau_q}$ kinds of data flow transmitted on *transmission link* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$ at time slot τ_q. Furthermore, denote $\mathcal{A}_{(i,j)}^{\tau_q}$ as the set of $N_{i,j}^{\tau_q}$ kinds of data flows going into node $v_j^{\tau_q}$ at time slot τ_q through *transmission link* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$, which is given by

$$\mathcal{A}_{(i,j)}^{\tau_q} = \Big\{\xi_n \mid n \in [N_{i,j}^{\tau_q}], v_i^{\tau_q} \text{ can transmit data flow } \xi_n \text{ to } v_j^{\tau_q}, \left(v_i^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},t}\Big\}. \quad (25)$$

Denote $\lambda_{i,j}^{\tau_q}(\xi_n)$ as the amount of data flow ξ_n transmitted on *transmission link* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$. Furthermore, denote $\lambda_{i,j}^{\tau_q}$ as the total amount of all data flows transmitted on *transmission link* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$. Therefore, $\lambda_{i,j}^{\tau_q}$ is equal to the total amount of $N_{i,j}^{\tau_q}$ kinds of data flow $\{\xi_n\}_{n\in\left[N_{i,j}^{\tau_q}\right]}$ transmitted on *transmission link* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$, which is given by [26]

$$\lambda_{i,j}^{\tau_q} = \sum_{n\in[N_{i,j}^{\tau_q}]} \lambda_{i,j}^{\tau_q}(\xi_n), \quad \forall \left(v_i^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},t}. \quad (26)$$

2) For Storage Link: Let $\lambda_j^{\tau_q,\tau_{q+1}}(\xi_n)$ denote the amount of data flow ξ_n stored on *storage links* $\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right)$. Denote $\lambda_j^{\tau_q,\tau_{q+1}}$ as the total amount of all data flows stored on *storage links* $\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right)$. There are two cases to consider:

1) If node $v_i^{\tau_q}$ is the non-computing function node in the MF-TEG, i.e., $v_i^{\tau_q} \in \mathcal{V}_{\mathcal{F},nc}$, then $N_i^{\tau_q}$ kinds of data flow $\{\xi_n\}_{n\in\left[N_i^{\tau_q}\right]}$ going into node $v_i^{\tau_q}$ is not processed. Therefore, only these $N_i^{\tau_q}$ kinds of data flow $\{\xi_n\}_{n\in\left[N_i^{\tau_q}\right]}$ go out of node $v_i^{\tau_q}$. Hence, $\lambda_j^{\tau_q,\tau_{q+1}}$ is equal to the total amount of $N_i^{\tau_q}$ kinds of data flow $\{\xi_n\}_{n\in\left[N_i^{\tau_q}\right]}$ stored on *storage links* $\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right)$, i.e.,

$$\lambda_j^{\tau_q,\tau_{q+1}} = \sum_{n\in[N_j^{\tau_q}]} \lambda_j^{\tau_q,\tau_{q+1}}(\xi_n), \quad \forall v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},nc}, \quad \left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right) \in \mathcal{L}_{\mathcal{F},s}. \quad (27)$$

2) If node $v_i^{\tau_q}$ is the sub-virtual node in the MF-TEG, i.e., $v_i^{\tau_q} \in \mathcal{V}_{\mathcal{F},sv}$, then M_i out of $N_i^{\tau_q}$ kinds of data flow $\{\xi_n\}_{n\in[N_i^{\tau_q}]}$ could be processed by M_i virtual computing nodes $\{v_{(i,c_m)}^{\tau_q}\}_{m\in[M_i]}$, and are transferred into M_i new kinds of data flows $\{\xi'_{c_m}\}_{m\in[M_i],c_m\in[N_i^{\tau_q}]}$. Furthermore, denote $\mathcal{B}_j^{\tau_q}$ as the set of different kinds of data flow by the *virtual transmission links* $\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right)$ $(\forall m \in [M_j])$ going into node $v_j^{\tau_q}$ at time slot τ_q, which is given by

$$\mathcal{B}_j^{\tau_q} = \Big\{\xi'_{c_m} \mid \text{ the data flow } \xi'_{c_m} \text{ going into the node } v_j^{\tau_q} \text{ through } \left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right), m \in [M_j], \left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},c}^{out}\Big\}. \quad (28)$$

Moreover, denote $\mathcal{D}_j^{\tau_q}$ as the set of different kinds of data flow going into node $v_j^{\tau_q}$ at time slot τ_q, which is given by

$$\mathcal{D}_j^{\tau_q} = \mathcal{A}_j^{\tau_q} \cup \mathcal{B}_j^{\tau_q}. \quad (29)$$

Denote $Z_j^{\tau_q}$ as the number of different kinds of data flows going into node $v_j^{\tau_q}$ at time slot τ_q. Hence, we have $Z_j^{\tau_q} = |\mathcal{D}_j^{\tau_q}|$. Therefore, $\lambda_j^{\tau_q,\tau_{q+1}}$ is equal to the total amount of $Z_i^{\tau_q}$ kinds of data flow $\{\xi_n\}_{n\in\left[Z_i^{\tau_q}\right]}$ stored on *storage links* $\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right)$, that is,

$$\lambda_j^{\tau_q,\tau_{q+1}} = \sum_{n\in[Z_j^{\tau_q}]} \lambda_j^{\tau_q,\tau_{q+1}}(\xi_n), \quad \forall v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},sv}, \quad \left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right) \in \mathcal{L}_{\mathcal{F},s}. \quad (30)$$

3) For Virtual Transmission Link: Denote $\lambda_{j,(j,c_m)}^{\tau_q}(\xi_{c_m})$ as the amount of data flow ξ_{c_m} transmitted on *virtual transmission link* $\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right)$. Denote $\lambda_{(j,c_m),j}^{\tau_q}(\xi'_{c_m})$ as the amount of data flow ξ'_{c_m} transmitted on *virtual transmission link* $\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right)$. Denote $\lambda_{j,(j,c_m)}^{\tau_q}$ and $\lambda_{(j,c_m),j}^{\tau_q}$ as the total amount of all data flows transmitted on *transmission links* $\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right)$ and $\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right)$, respectively. For the *virtual transmission link* $\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right)$, only data flow ξ_{c_m} is allowed to use. Therefore, $\lambda_{j,(j,c_m)}^{\tau_q}$ is equal to the amount of data flow ξ_{c_m} transmitted on *virtual transmission link* $\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right)$, i.e.,

$$\lambda_{j,(j,c_m)}^{\tau_q} = \lambda_{j,(j,c_m)}^{\tau_q}(\xi_{c_m}), \quad \forall \left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},c}^{in}. \quad (31)$$

Furthermore, the data flow ξ_{c_m} is processed by computing function $f_{\xi_{c_m},\xi'_{c_m}}$ at virtual computing function node $v_{(j,c_m)}^{\tau_q}$ and is transformed into a new kind of data flow ξ'_{c_m}. As the output of the virtual computing function node $v_{(j,c_m)}^{\tau_q}$ is only data flow ξ'_{c_m}, $\lambda_{(j,c_m),j}^{\tau_q}$ is equal to the amount of data flow ξ'_{c_m} transmitted on *virtual transmission link* $\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right)$, i.e.,

$$\lambda_{(j,c_m),j}^{\tau_q} = \lambda_{(j,c_m),j}^{\tau_q}(\xi'_{c_m}), \quad \forall \left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},c}^{out}. \quad (32)$$

C. Fundamental Constrains

There are four kinds of fundamental constraints in MF-TEG: computation capacity constraints, communication capacity constraints, storage capacity constraints, and flow conservation constraints.

1) Computation Capacity Constraints: Denote $\alpha_{\xi_{c_m},\xi'_{c_m}}$ as the computation factor for transferring the data flow ξ_{c_m} into a new kind of data flow ξ'_{c_m} is $\alpha_{\xi_{c_m},\xi'_{c_m}}$, with processing units per data flow unit (for example, an image has a fixed amount of data and processing an image needs a fixed number of flops operations) [24], [25], [26]. Furthermore, denote the computation capacity of the virtual computing function node $v_{(j,c_m)}^{\tau_q}$ as $P_{j,c_m}^{\tau_q}$, which represents the maximum amount of data processing capability for fulfilling the function $f_{\xi_{c_m},\xi'_{c_m}}$ at the virtual computing node $v_{(j,c_m)}^{\tau_q}$ during the timeslot τ_q (for example, the number of flops operation virtual computing function node $v_{(j,c_m)}^{\tau_q}$ can provide during timeslot τ_q) [25], [26].

Hence, the data processing capability for fulfilling the function $f_{\xi_{c_m},\xi'_{c_m}}$ at the virtual computing node $v_{(j,c_m)}^{\tau_q}$ is upper bounded by $P_{j,c_m}^{\tau_q}$ [25], [26], i.e.,

$$\lambda_{j,(j,c_m)}^{\tau_q}(\xi_{c_m}) \cdot \alpha_{\xi_{c_m},\xi'_{c_m}} \leq P_{j,c_m}^{\tau_q}, \quad \forall v_{(j,c_m)}^{\tau_q} \in \mathcal{V}_{\mathcal{F},c}. \quad (33)$$

2) Communication Capacity Constraints: There are three cases to consider.

1) For the *transmission link* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$, the total amount of all data flows transmitted on *transmission link* $\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$ can not exceed communication capacity $C_{\mathcal{F},t}\left(v_i^{\tau_q}, v_j^{\tau_q}\right)$ [20], [24], [26], [34], i.e.,

$$0 \leq \lambda_{i,j}^{\tau_q} \leq C_{\mathcal{F},t}\left(v_i^{\tau_q}, v_j^{\tau_q}\right), \quad \forall\left(v_i^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},t}. \tag{34}$$

2) For the *virtual transmission link* $\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right)$, the total amount of the data flow ξ_n transmitted on it cannot exceed its communication capacity for ξ_n, i.e.,

$$\begin{aligned} 0 \leq \lambda_{j,(j,c_m)}^{\tau_q}(\xi_n) \leq C_{\mathcal{F},\xi_n}\left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right), \\ \forall \xi_n, n \in \left[N_j^{\tau_q}\right], \left(v_j^{\tau_q}, v_{(j,c_m)}^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},c}^{in}. \end{aligned} \tag{35}$$

3) For the *virtual transmission link* $\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right)$, the total amount of the data flow ξ_n' transmitted on it cannot exceed its communication capacity for the data flow ξ_n', i.e.,

$$\begin{aligned} 0 \leq \lambda_{(j,c_m),j}^{\tau_q}(\xi_n') \leq C_{\mathcal{F},\xi_n'}\left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right), \\ \forall \xi_n', n \in \left[N_j^{\tau_q}\right], \left(v_{(j,c_m)}^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},c}^{out}. \end{aligned} \tag{36}$$

3) Storage Capacity Constraints: The total amount of all data flows stored on *storage links* $\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right)$ is upper bounded by the storage capacity $C_{\mathcal{F},s}\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right)$ [20], [33], [34], i.e.,

$$0 \leq \lambda_j^{\tau_q,\tau_{q+1}} \leq C_{\mathcal{F},s}\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right), \quad \forall\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right) \in \mathcal{L}_{\mathcal{F},s}. \tag{37}$$

4) Flow Conservation Constraints: there are three cases to consider:

1) If node $v_j^{\tau_q}$ is the non-computing function node in the MF-TEG, i.e., $v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},nc}$, then the data flow $\xi_n\left(\forall n \in [N_j^{\tau_q}]\right)$ going into this node $v_j^{\tau_q}$ is not processed. Therefore, during timeslot τ_q, the total amount of the data flow $\xi_n\left(\forall n \in [N_j^{\tau_q}]\right)$ going into the node $v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},nc}$ is equal to the total amount of the data flow ξ_n coming out of this node $v_j^{\tau_q}$, i.e., [20], [26]

$$\begin{aligned} \sum_{v_i^{\tau_q}:(v_i^{\tau_q}, v_j^{\tau_q}) \in \mathcal{L}_{\mathcal{F},t}} \lambda_{i,j}^{\tau_q}(\xi_n) + \lambda_j^{\tau_{q-1},\tau_q}(\xi_n) \\ = \sum_{v_i^{\tau_q}:(v_j^{\tau_q}, v_i^{\tau_q}) \in \mathcal{L}_{\mathcal{F},t}} \lambda_{j,i}^{\tau_q}(\xi_n) + \lambda_j^{\tau_q,\tau_{q+1}}(\xi_n), \\ \forall v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},nc}, n \in [N_j^{\tau_q}]. \end{aligned} \tag{38}$$

2) If node $v_j^{\tau_q}$ is the sub-virtual node in the MF-TEG, i.e., $v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},sv}$, then the data flow $\xi_n\left(\forall \xi_n \in \mathcal{D}_j^{\tau_q}\right)$ going into this node $v_j^{\tau_q}$ is not processed. Therefore, during timeslot τ_q, the total amount of the data flow $\xi_n\left(\forall \xi_n \in \mathcal{D}_j^{\tau_q}\right)$ going into the node $v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},sv}$ is equal to the total amount of the data flow $\xi_n\left(\forall \xi_n \in \mathcal{D}_j^{\tau_q}\right)$ coming out of this node $v_j^{\tau_q}$, i.e., [20], [26]

$$\begin{aligned} \lambda_j^{\tau_{q-1},\tau_q}(\xi_n) + \sum_{v_i^{\tau_q}:(v_i^{\tau_q}, v_j^{\tau_q}) \in \mathcal{L}_{\mathcal{F},t}} \lambda_{i,j}^{\tau_q}(\xi_n) \\ + \sum_{v_{j,c_m}^{\tau_q}:(v_{j,c_m}^{\tau_q}, v_j^{\tau_q}) \in \mathcal{L}_{\mathcal{F},c}^{in}} \lambda_{(j,c_m),j}^{\tau_q}(\xi_n) \\ = \lambda_j^{\tau_q,\tau_{q+1}}(\xi_n) + \sum_{v_i^{\tau_q}:(v_j^{\tau_q}, v_i^{\tau_q}) \in \mathcal{L}_{\mathcal{F},t}} \lambda_{j,i}^{\tau_q}(\xi_n) \\ + \sum_{v_{j,c_m}^{\tau_q}:(v_j^{\tau_q}, v_{j,c_m}^{\tau_q}) \in \mathcal{L}_{\mathcal{F},c}^{out}} \lambda_{j,(j,c_m)}^{\tau_q}(\xi_n), \\ \forall v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},sv}, \forall \xi_n \in \mathcal{D}_j^{\tau_q}. \end{aligned} \tag{39}$$

3) If the node is the virtual computing function node $v_{(j,c_m)}^{\tau_q}$ in the MF-TEG, i.e., $v_{(j,c_m)}^{\tau_q} \in \mathcal{V}_{\mathcal{F},pu}$, then the data flow ξ_{c_m} going into the virtual computing function node $v_{(j,c_m)}^{\tau_q}$ is processed and transferred into a new kind of data flow ξ_{c_m}' coming out of this virtual computing function node. Denote $\beta_{\xi_{c_m},\xi_{c_m}'}$ as a scaling factor that represents the ratio between data flow ξ_{c_m} and ξ_{c_m}' (for example, an image is compressed into a new image) [25]. Hence, we have

$$\begin{aligned} \lambda_{(j,c_m),j}^{\tau_q}(\xi_{c_m}') = \beta_{\xi_{c_m},\xi_{c_m}'} \cdot \lambda_{j,(j,c_m)}^{\tau_q}(\xi_{c_m}), \\ \forall v_{j,c_m}^{\tau_q} \in \mathcal{V}_{\mathcal{F},c}. \end{aligned} \tag{40}$$

IV. Exemplary Application

In this section, we provide an exemplary application of SAGIN aided multi-tier computing network which consists of four satellites, one ground station, one device, and one cloud computing service, as shown in Fig. 7. In SAGIN aided multi-tier computing network, the device offloads its mission flow to the cloud computing server through SN [2], [13]. Specifically, the device offloads its mission flow to one of its nearby satellites. Furthermore, this satellite can transmit this mission flow to ground stations through the SN [2], [13]. Moreover, the ground station forward this mission flow to the cloud computing server via cables. Therefore, this nearby satellite and the cloud computing server can be treated as the source node v_s and the destination node v_d of this mission flow, respectively.

Moreover, suppose that the mission flow from the source node v_s to the destination node v_d requests a sequence of specific services. The requested services can be given by the SFC, consisting of a set of virtual network functions (VNFs) that have to be performed in the predefined order [20], [24], [26]. Denote the SFC of this data flow as $\mathcal{F} = \{f_1 \rightarrow f_2 \rightarrow f_3\}$, where f_l $(\forall 1 \leq l \leq 3)$ represents the lth function received by the data flow. Therefore, during the transmission in SN, this mission flow invokes one SFC [20], [31]. Furthermore, each function requires computing capability. In this case, for SN with SFC, satellite nodes deployed with VNFs provide communication, storage, and computing function. Moreover, both functions f_1 and f_2 are provided by the node v_1, while

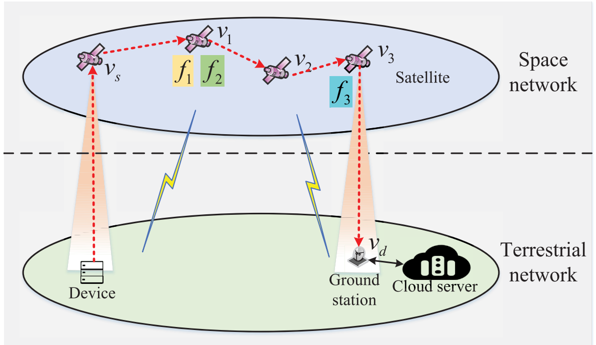


Fig. 7. An example SAGIN aided multi-tier computing network with four satellites, one device and one cloud service.

the function f_3 is provided by the node v_3. As the mission flow receives the two functions in one same node, the traditional TEG is not applicable. Hence, we adopt the MF-TEG to model the dynamic but predictable topology evolution of SN and to characterize the communication, storage, and computing capability of each node in the SN, as shown in Fig. 8(b).

A. MF-TEG Model for SN With SFC

Denote the MF-TEG in Fig. 8(b) as $\mathcal{G}_{\mathcal{F}}=(\mathcal{V}_{\mathcal{F}},\mathcal{L}_{\mathcal{F}})$, where $\mathcal{V}_{\mathcal{F}}$ and $\mathcal{L}_{\mathcal{F}}$ represent the set of nodes and links in the MF-TEG, respectively.

- **i) Vertices:** In this example, the set of non-computing function nodes, sub-virtual nodes, and virtual computing nodes can be expressed as $\mathcal{V}_{\mathcal{F},nc}=\{v_s^{\tau_q}\mid 1\le q\le 3\}\cup\{v_2^{\tau_q}\mid 1\le q\le 3\}\cup\{v_d^{\tau_q}\mid 1\le q\le 3\}$, $\mathcal{V}_{\mathcal{F},sv}=\{v_1^{\tau_q}\mid 1\le q\le 3\}\cup\{v_3^{\tau_q}\mid 1\le q\le 3\}$, and $\mathcal{V}_{\mathcal{F},c}=\{v_{(1,c_{f_1})}^{\tau_q}\mid 1\le q\le 3\}\cup\{v_{(1,c_{f_2})}^{\tau_q}\mid 1\le q\le 3\}\cup\{v_{(3,c_{f_3})}^{\tau_q}\mid 1\le q\le 3\}$, respectively. Furthermore, the virtual computing nodes $v_{(1,c_{f_1})}^{\tau_q}(\forall 1\le q\le 3)$, $v_{(1,c_{f_2})}^{\tau_q}(\forall 1\le q\le 3)$, and $v_{(3,c_{f_2})}^{\tau_q}(\forall 1\le q\le 3)$ can provide functions f_1, f_2, and f_3 for the mission flow, respectively.
- **ii) Links:** In this example, the set of *transmission links*, *storage links*, and *virtual transmission links* can be expressed as $\mathcal{L}_{\mathcal{F},t}=\{(v_i^{\tau_q},v_j^{\tau_q})\mid 1\le q\le 3, v_i$ can transmit data flows to v_j during timeslot $\tau_q\}$, $\mathcal{L}_{\mathcal{F},s}=\left\{(v_j^{\tau_q},v_j^{\tau_{q+1}})\mid v_j\in\{v_s,v_1,v_2,v_3,v_d\},1\le q\le 2\right\}$, and $\mathcal{L}_{\mathcal{F},c}=\mathcal{L}_{\mathcal{F},c}^{in}\cup\mathcal{L}_{\mathcal{F},c}^{out}$, respectively, where $\mathcal{L}_{\mathcal{F},c}^{in}=\left\{(v_1^{\tau_q},v_{(1,c_{f_1})}^{\tau_q})\mid 1\le q\le 3\right\}\{(v_1^{\tau_q},v_{(1,c_{f_2})}^{\tau_q})\mid 1\le q\le 3\}\cup\{(v_3^{\tau_q},v_{(3,c_{f_3})}^{\tau_q})\mid 1\le q\le 3\right\}$ and $\mathcal{L}_{\mathcal{F},c}^{out}=\left\{(v_{(1,c_{f_1})}^{\tau_q},v_1^{\tau_q})\mid 1\le q\le 3\right\}\cup\left\{(v_{(1,c_{f_2})}^{\tau_q},v_1^{\tau_q})\mid 1\le q\le 3\right\}\cup\left\{(v_{(3,c_{f_2})}^{\tau_q},v_3^{\tau_q})\mid 1\le q\le 3\right\}$.

The data flow just coming out of a source node $v_s^{\tau_q}$ without receiving any function is denoted as ξ_{f_0}, and the data flow just receiving function $f_l\in\mathcal{F}$ is denoted as ξ_{f_l} [20], [26]. Furthermore, the data flow $\xi_{f_{l-1}}$ going into the virtual computing node $v_{(j,c_{f_l})}^{\tau_q}(\forall f_l\in\mathcal{F},1\le q\le 3)$ is processed by function f_l and it is transformed into a new kind of data flow ξ_{f_l}. And the output of the virtual computing node $v_{(j,c_{f_l})}^{\tau_q}$ is the data flow ξ_{f_l}.

Denote $C_{\mathcal{F},\xi_{f_l}}\left(v_j^{\tau_q},v_{(j,c_{f_n})}^{\tau_q}\right)$ and $C_{\mathcal{F},\xi_{f_l}}\left(v_{(j,c_{f_n})}^{\tau_q},v_j^{\tau_q}\right)$ as the communication capacity of *virtual transmission links* $\left(v_j^{\tau_q},v_{(j,c_{f_n})}^{\tau_q}\right)$ and $\left(v_{(j,c_{f_n})}^{\tau_q},v_j^{\tau_q}\right)$ $(\forall f_n\in\mathcal{F})$ for the data flow ξ_{f_l} $(\forall f_l\in\mathcal{F}\cup\{f_0\})$ in the MF-TEG, respectively. Moreover, as the virtual computing node $v_{(j,c_{f_n})}^{\tau_q}$ only provides the computing function f_n, only data flow $\xi_{f_{n-1}}$ is allowed to use *virtual transmission link* $\left(v_j^{\tau_q},v_{(j,c_{f_n})}^{\tau_q}\right)$, while only data flow ξ_{f_n} is allowed to use the *virtual transmission link* $\left(v_{(j,c_{f_n})}^{\tau_q},v_j^{\tau_q}\right)$. Consequently, the communication capacity $C_{\mathcal{F},\xi_{f_l}}\left(v_j^{\tau_q},v_{(j,c_{f_n})}^{\tau_q}\right)$ and $C_{\mathcal{F},\xi_{f_l}}\left(v_{(j,c_{f_n})}^{\tau_q},v_j^{\tau_q}\right)$ are given by

$$C_{\mathcal{F},\xi_{f_l}}\left(v_j^{\tau_q},v_{(j,c_{f_n})}^{\tau_q}\right)=\begin{cases}\infty, & f_l=f_{n-1},\\ 0, & f_l\neq f_{n-1},\end{cases}\ \forall\left(v_j^{\tau_q},v_{(j,c_{f_n})}^{\tau_q}\right)\in\mathcal{L}_{\mathcal{F},c}^{in}, \tag{41}$$

$$C_{\mathcal{F},\xi_{f_l}}\left(v_{(j,c_{f_n})}^{\tau_q},v_j^{\tau_q}\right)=\begin{cases}\infty, & f_l=f_n,\\ 0, & f_l\neq f_n,\end{cases}\ \forall\left(v_{(j,c_{f_n})}^{\tau_q},v_j^{\tau_q}\right)\in\mathcal{L}_{\mathcal{F},c}^{out}. \tag{42}$$

Denote $\lambda_{i,j}^{\tau_q}(\xi_{f_l})$ (in Gbit) as the amount of data flow $\xi_{f_l}\,(\forall f_l\in\mathcal{F}\cup\{f_0\})$ transmitted on *transmission link* $\left(v_i^{\tau_q},v_j^{\tau_q}\right)$. Let $\lambda_j^{\tau_q,\tau_{q+1}}(\xi_{f_l})$ (in Gbit) denote the amount of data flow $\xi_{f_l}\,(\forall f_l\in\mathcal{F}\cup\{f_0\})$ stored on *storage links* $\left(v_j^{\tau_q},v_j^{\tau_{q+1}}\right)$. Furthermore, denote $\lambda_{j,(j,c_{f_n})}^{\tau_q}(\xi_{f_l})$ and $\lambda_{(j,c_{f_n}),j}^{\tau_q}(\xi_{f_l})$ (in Gbit) as the amount of data flow $\xi_{f_l}\,(\forall f_l\in\mathcal{F}\cup\{f_0\})$ transmitted on *virtual transmission link* $\left(v_j^{\tau_q},v_{(j,c_{f_n})}^{\tau_q}\right)$ and $\left(v_{(j,c_{f_n})}^{\tau_q},v_j^{\tau_q}\right)$, respectively.

Denote $\alpha_{\xi_{f_{l-1}},\xi_{f_l}}$ as the computation factor for transferring the data flow $\xi_{f_{l-1}}$ into a new kind of data flow ξ_{f_l}, with processing units per data flow unit [25]. Furthermore, denote the computation capacity of the virtual computing function node $v_{(j,c_{f_l})}^{\tau_q}$ as $P_{j,c_{f_l}}^{\tau_q}$, which represents the maximum amount of data processing capability for fulfilling the function f_l at the virtual computing node $v_{(j,c_{f_l})}^{\tau_q}$ during the timeslot τ_q.

B. Fundamental Constrains

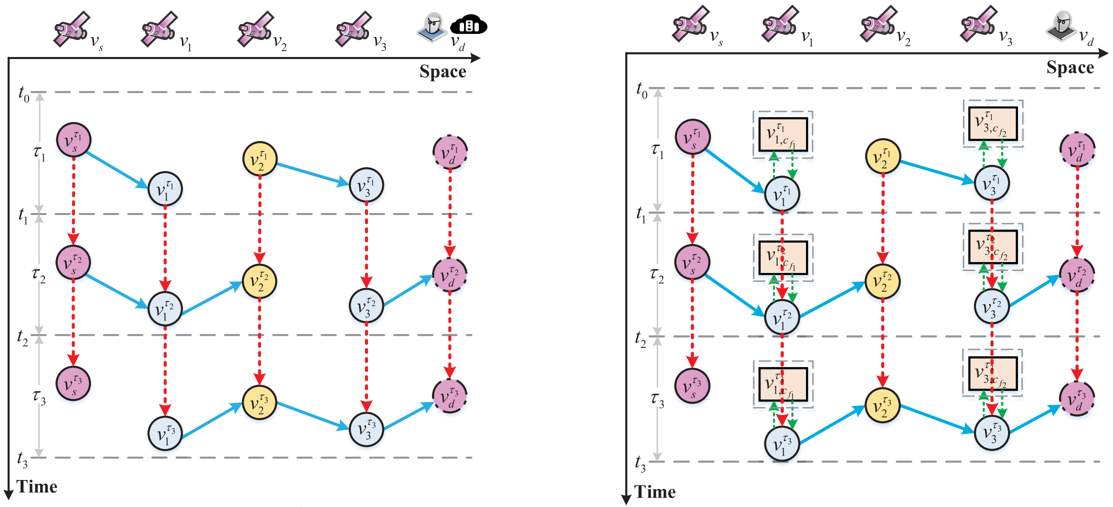
1) Computation Capacity Constraints: For the virtual computing function node $v_{(j,c_{f_l})}^{\tau_q}\in\mathcal{V}_{\mathcal{F},c}$, the data processing capability for fulfilling the function f_l at the virtual computing node $v_{(j,c_{f_l})}^{\tau_q}$ is upper bounded by $P_{j,c_{f_l}}^{\tau_q}$ [25], [26]. According to (33), we have

$$\lambda_{j,(j,c_{f_l})}^{\tau_q}(\xi_{f_{l-1}})\cdot\alpha_{\xi_{f_{l-1}},\xi_{f_l}}\le P_{j,c_{f_l}}^{\tau_q},\quad\forall v_{(j,c_{f_l})}^{\tau_q}\in\mathcal{V}_{\mathcal{F},c}. \tag{43}$$

2) Communication Capacity Constraints: The total amount of all data flows transmitted on a link cannot exceed its communication capacity [20], [24], [26], [34]. According to (34)-(36), we have

$$0\le\sum_{f_l\in\mathcal{F}\cup\{f_0\}}\lambda_{i,j}^{\tau_q}(\xi_{f_l})\le C_{\mathcal{F},t}\left(v_i^{\tau_q},v_j^{\tau_q}\right),\quad\forall\left(v_i^{\tau_q},v_j^{\tau_q}\right)\in\mathcal{L}_{\mathcal{F},t}, \tag{44}$$

$$0\le\lambda_{j,(j,c_{f_n})}^{\tau_q}(\xi_{f_l})\le C_{\mathcal{F},\xi_{f_l}}\left(v_j^{\tau_q},v_{(j,c_{f_n})}^{\tau_q}\right),\quad\forall f_l\in\mathcal{F}\cup\{f_0\},\left(v_j^{\tau_q},v_{(j,c_{f_n})}^{\tau_q}\right)\in\mathcal{L}_{\mathcal{F},c}^{in}, \tag{45}$$



(a) The time expanded graph for the example SN

(b) The multi-functional time expanded graph for the example SN

Fig. 8. Topology graph for the example SN with three timeslots.

$$0 \leq \lambda_{(j,c_{f_n}),j}^{\tau_q}(\xi_{f_l}) \leq C_{\mathcal{F},\xi_{f_l}}\left(v_{(j,c_{f_n})}^{\tau_q}, v_j^{\tau_q}\right),$$
$$\forall f_l \in \mathcal{F} \cup \{f_0\}, \left(v_{(j,c_{f_n})}^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},c}^{out}. \tag{46}$$

3) Storage Capacity Constraints: According to (37), we have

$$0 \leq \sum\nolimits_{f_l \in \mathcal{F} \cup \{f_0\}} \lambda_j^{\tau_q,\tau_{q+1}}(\xi_{f_l}) \leq C_{\mathcal{F},s}\left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right),$$
$$\forall \left(v_j^{\tau_q}, v_j^{\tau_{q+1}}\right) \in \mathcal{L}_{\mathcal{F},s}. \tag{47}$$

4) The Flow Conservation Constraints: there are two cases to consider:

1) If node $v_j^{\tau_q}$ is the non-computing function node in the MF-TEG, i.e., $v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},nc}$, then the data flow ξ_{f_l} $(\forall f_l \in \mathcal{F} \cup \{f_0\})$ going into this node $v_j^{\tau_q}$ is not processed. Therefore, during timeslot τ_q, the total amount of the data flow ξ_{f_l} going into the node $v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},nc}$ is equal to the total amount of the data flow ξ_{f_l} coming out of this node $v_j^{\tau_q}$ [20], [26]. According to (38), we have

$$\sum_{v_i^{\tau_q}:(v_i^{\tau_q},v_j^{\tau_q}) \in \mathcal{L}_{\mathcal{F},t}} \lambda_{i,j}^{\tau_q}(\xi_{f_l}) + \lambda_j^{\tau_{q-1},\tau_q}(\xi_{f_l})$$
$$= \sum_{v_i^{\tau_q}:(v_j^{\tau_q},v_i^{\tau_q}) \in \mathcal{L}_{\mathcal{F},t}} \lambda_{j,i}^{\tau_q}(\xi_{f_l}) + \lambda_j^{\tau_q,\tau_{q+1}}(\xi_{f_l}),$$
$$\forall v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},nc}, f_l \in \mathcal{F} \cup \{f_0\}. \tag{48}$$

2) If node $v_j^{\tau_q}$ is the sub-virtual node in the MF-TEG, i.e., $v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},sv}$, then node $v_j^{\tau_q}$ acts only as a relay for the data flow. Therefore, the data flow ξ_{f_l} $(\forall f_l \in \mathcal{F} \cup \{f_0\})$ going into this node $v_j^{\tau_q}$ is not processed. Hence, during timeslot τ_q, the total amount of the data flow ξ_{f_l} going into the node $v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},sv}$ is equal to the total amount of the data flow ξ_{f_l} coming out of this node $v_j^{\tau_q}$ [20], [26]. According to (39), we have

$$\sum_{v_i^{\tau_q}:(v_i^{\tau_q},v_j^{\tau_q}) \in \mathcal{L}_{\mathcal{F},t}} \lambda_{i,j}^{\tau_q}(\xi_{f_l}) + \lambda_j^{\tau_{q-1},\tau_q}(\xi_{f_l})$$
$$+ \sum_{v_{j,c_{f_n}}^{\tau_q}:(v_{j,c_{f_n}}^{\tau_q},v_j^{\tau_q}) \in \mathcal{L}_{\mathcal{F},c}^{in}} \lambda_{(j,c_{f_n}),j}^{\tau_q}(\xi_{f_l})$$
$$= \sum_{v_i^{\tau_q}:(v_j^{\tau_q},v_i^{\tau_q}) \in \mathcal{L}_{\mathcal{F},t}} \lambda_{j,i}^{\tau_q}(\xi_{f_l}) + \lambda_j^{\tau_q,\tau_{q+1}}(\xi_{f_l}),$$
$$+ \sum_{v_{j,c_{f_n}}^{\tau_q}:(v_j^{\tau_q},v_{j,c_{f_n}}^{\tau_q}) \in \mathcal{L}_{\mathcal{F},c}^{out}} \lambda_{j,(j,c_{f_n})}^{\tau_q}(\xi_{f_l}),$$
$$\forall v_j^{\tau_q} \in \mathcal{V}_{\mathcal{F},sv}, f_l \in \mathcal{F} \cup \{f_0\}. \tag{49}$$

3) If the node is the virtual computing node $v_{(i,c_{f_l})}^{\tau_q}$ in the MF-TEG, i.e., $v_{(i,c_{f_l})}^{\tau_q} \in \mathcal{V}_{\mathcal{F},c}$, then the data flow $\xi_{f_{l-1}}$ going into the virtual computing node $v_{(i,pu_{f_l})}^{\tau_q}$ is processed and transferred into the data flow ξ_{f_l} coming out of this virtual computing node. Denote $\beta_{\xi_{f_{l-1}},\xi_{f_l}}$ as a scaling factor that represents the ratio between data flow $\xi_{f_{l-1}}$ and ξ_{f_l} [25]. According to (40), we have

$$\lambda_{(j,c_{f_l}),j}^{\tau_q}(\xi_{f_l}) = \beta_{\xi_{f_{l-1}},\xi_{f_l}} \cdot \lambda_{j,(j,c_{f_l})}^{\tau_q}(\xi_{f_{l-1}}),$$
$$\forall v_{j,c_{f_l}}^{\tau_q} \in \mathcal{V}_{\mathcal{F},c}. \tag{50}$$

5) The Demand Satisfaction Constraints: Let λ_s (in Gbit) denote the total amount of the data flow coming out of all the replicas of the source node during the time horizon $\mathcal{T}$. As the replica $v_s^{\tau_q}$ of the source node cannot provide any function for the data flow, the data flow coming out of $v_s^{\tau_q}$ is only ξ_{f_0} [20]. Thus, we have

$$\lambda_s = \sum_{q=1}^{3} \sum_{v_j^{\tau_q}:(v_s^{\tau_q},v_j^{\tau_q}) \in \mathcal{L}_{\mathcal{F},t}} \lambda_{s,j}^{\tau_q}(\xi_{f_0}). \tag{51}$$

Let λ_d (in Gbit) denote the total amount of the data flow going into all the replicas of the destination node during the time horizon $\mathcal{T}$. In order to complete the requested service, the data flow must satisfy the SFC constraints, i.e., the data flow from the source node to the destination node should be processed by functions in the predefined order [20]. Therefore, the data flow going into $v_d^{\tau_q}$ is only ξ_{f_2}, we have

$$\lambda_d = \sum_{q=1}^{3} \sum_{v_j^{\tau_q}:\left(v_j^{\tau_q}, v_d^{\tau_q}\right)\in\mathcal{L}_{\mathcal{F},t}} \lambda_{j,d}^{\tau_q}\left(\xi_{f_2}\right). \tag{52}$$

Furthermore, the scaling factor between λ_s and λ_d is $\beta_{\xi_{f_0},\xi_{f_1}} \cdot \beta_{\xi_{f_1},\xi_{f_2}}$, i.e.,

$$\lambda_d = \beta_{\xi_{f_0},\xi_{f_1}} \cdot \beta_{\xi_{f_1},\xi_{f_2}} \cdot \lambda_s. \tag{53}$$

C. Maximum Flow Problem

In this application, we want to maximize the amount of data flow going into all the replicas of the destination node during the time horizon $\mathcal{T}$. Therefore, this problem can be formulated as follows [20]: $\mathbf{P1}$, as shown at the bottom of the page, which is a linear programming (LP) problem, as both the objective and constraints are linear functions, and can be solved using the existing solver, for example, CVX, Cplex, Lingo, and so on.

Remark 2: Although this example of the SAGIN aided multi-tier computing network only consists of four satellites, one ground station, one device, and one cloud computing service, the same philosophy is valid for SN with different system configurations. Hence, MF-TEG modelling is applicable to different scales of SN with SFC.

V. Complexity Analysis

In this section, we analyze the time complexity of solving the problem based on MF-TEG by LP algorithm and that of solving the problem based on SSG-aided VNG by LP algorithm. Assume that each computing function node can provide at most M computing functions. Denote K as the total number of data flows in the network. We consider the worst case, where each node is connected to each other in each timeslot and all nodes are computing function nodes, which can provide M computing functions.

Theorem 1: The time complexity of solving the problem based on MF-TEG by LP algorithm is approximately $\mathcal{O}\left(\left(KT\left(N^2+2MN\right)\right)^{3.5}\right)$.

Proof of Theorem 1: In this case, the number of transmission links and virtual transmission links in each timeslot are $N(N-1)$ and $2NM$, respectively. Therefore, the total number of transmission links and virtual transmission links in MF-TEG are $N(N-1)T$ and $2NMT$, respectively. Furthermore, the total number of storage links in MF-TEG is $N(T-1)$. Therefore, the total number of links in MF-TEG is $N\left(N-1\right)T+2NMT+N\left(T-1\right)=N\left(NT+2MT-1\right)$. Moreover, there are at most K data flows on each link. Therefore, the total number of variables in the problem based on MF-TEG is $KN\left(NT+2MT-1\right)$. Hence, The time complexity of solving the problem based on MF-TEG by LP algorithm is $\mathcal{O}\left(\left(KN\left(NT+2MT-1\right)\right)^{3.5}\right) \approx \mathcal{O}\left(\left(KT\left(N^2+2MN\right)\right)^{3.5}\right)$ [35]. ■

Theorem 2: The time complexity of solving the problem based on SSG-aided VNG by LP algorithm is approximately $\mathcal{O}\left(T\left(\left(K\left(N^2+2NM\right)\right)^{3.5}\right)\right)$.

Proof of Theorem 2: The total number of transmission links and virtual transmission links in SSG-aided VNG for each timeslot are $N(N-1)$ and $2NM$, respectively. Therefore, the total number of links in SSG-aided VNG is $N\left(N-1\right)+2NM$. Moreover, there are at most K data flows on each link. Therefore, the total number of variables in the problem based on SSG-aided VNG for each timeslot is $K\left(N^2+2NM-N\right)$. Hence, The time complexity of solving the problem based on SSG-aided VNG by LP algorithm for each timeslot is $\mathcal{O}\left(\left(K\left(N^2+2NM-1\right)\right)^{3.5}\right)$ [35]. As there are T timeslots in time horizon, the time complexity of solving the problem based on SSG-aided VNG by LP algorithm is $\mathcal{O}\left(T\left(\left(K\left(N^2+2NM-1\right)\right)^{3.5}\right)\right) \approx \mathcal{O}\left(T\left(\left(K\left(N^2+2NM\right)\right)^{3.5}\right)\right)$ ■

Remark 3: According to theorem 1 and theorem 2, the time complexity of solving the problem based on MF-TEG by LP algorithm is at least $T^{2.5}$-fold compared with that of solving the problem based on SSG-aided VNG by LP algorithm.

VI. Simulation Results

In this section, we demonstrate the performance of the MF-TEG modelling for joint communication, computation, and storage under the scenario of SAGIN aided multi-tier computing for SN with SFC in Section IV. Furthermore, we use SSG-aided VNG (SSG-VNG) as the benchmark, where the SSG is used to describe the dynamic evolution of the

$$\begin{aligned}
\mathbf{P1}: \quad & \max_{\substack{\left\{\lambda_{i,j}^{\tau_q}\left(\xi_{f_l}\right)\right\},\ \left\{\lambda_j^{\tau_q,\tau_q+1}\left(\xi_{f_l}\right)\right\} \\ \left\{\lambda_{j,(j,c_{f_n})}^{\tau_q}\left(\xi_{f_l}\right)\right\},\ \left\{\lambda_{(j,c_{f_n}),j}^{\tau_q}\left(\xi_{f_l}\right)\right\}}} \quad \lambda_d \\
& \text{s.t.} \quad (43)-(53) \\
& \qquad \lambda_{i,j}^{\tau_q}\left(\xi_{f_l}\right) \geq 0, \forall\left(v_i^{\tau_q}, v_j^{\tau_q}\right) \in \mathcal{L}_{\mathcal{F},t}, f_l \in \mathcal{F} \cup \{f_0\}, \\
& \qquad \lambda_j^{\tau_q,\tau_q+1}\left(\xi_{f_l}\right) \geq 0, \forall\left(v_j^{\tau_q}, v_j^{\tau_q+1}\right) \in \mathcal{L}_{\mathcal{F},s}, f_l \in \mathcal{F} \cup \{f_0\}, \\
& \qquad \lambda_{j,(j,c_{f_n})}^{\tau_q}\left(\xi_{f_l}\right) \geq 0, \forall(v_j^{\tau_q}, v_{(j,c_{f_n})}^{\tau_q}) \in \mathcal{L}_{\mathcal{F},c}^{in}, f_l \in \mathcal{F} \cup \{f_0\}, \\
& \qquad \lambda_{(j,pu),j}^{\tau_q}\left(\xi_{f_l}\right) \geq 0, \forall v_{(j,pu)}^{\tau_q}, v_j^{\tau_q}) \in \mathcal{L}_{\mathcal{F},c}^{out}, f_l \in \mathcal{F} \cup \{f_0\}.
\end{aligned}$$

topology through the snapshots in discrete timeslots without considering the connection among snapshots, and for each snapshot, the VNG in [24] is adopted, which can only jointly model for communication and computation. Both of them are formulated as LP problem, which is solved by the optimization solver Gurobi 7.52 [36].

Simulations are conducted on an SN scenario which consists of 12 satellites, four ground stations, and one cloud computing server. Specifically, 12 satellites are arbitrarily selected from the Iridium Next constellation [37], and are distributed over six polar orbits at a height of 781 km. Four ground stations are located at Sanya, China (18°N, 109.5°E), Kashi, China (39.5°N, 76°E), Beijing, China (40°N, 116°E), and Xi'an, China (34°N, 108°E). Moreover, the cloud computing server is set at Xi'an. The time-varying network topology of the SN scenario is generated by the Satellite Tool Kit (STK) simulator. The time horizon $\mathcal{T}$ is set to 60 minutes.

In the SN scenario, the device offloads its mission flow to a nearby satellite, which can be considered as a source node. The source node v_s is randomly chosen from one of the satellites, while the destination node v_d is the cloud computing server located at Xi'an, China (34°N, 108°E). We set the length of SFC to 3, i.e., the SFC $\mathcal{F} = \{f_1 \rightarrow f_2 \rightarrow f_3\}$ consists of three functions. Furthermore, the two functions f_1 and f_2 are provided by one function node, while the function f_3 is provided by another function node. In this case, the traditional TEG cannot model the flow conservation for the coupled functions in one same node, hence not applicable. Moreover, we randomly select two nodes from all the nodes except v_s and v_d as function nodes.

In addition, the bandwidth of the satellite-ground link and the inter-satellite link are randomly chosen from [50,300] Mbps and [100,1000] Mbps, respectively [38]. The storage capacity of each satellite is set to be 500 Gbits, unless otherwise specified. The computational capability of function nodes is set to be 200 units, unless otherwise specified. Furthermore, we set the computation factor $\alpha_{\xi_{f_{l-1}},\xi_{f_l}} = 2$ $(\forall 1 \leq l \leq 3)$. In simulations, the results are obtained by averaging over 100 times simulation.

Firstly, we investigate the impact of the storage capacity on the network maximum flow. In Fig. 9, the network maximum flow versus the storage capacity is plotted. It can be observed that with the increasing storage capacity, the network maximum flow in SSG-VNG stays at one constant level, while the network maximum flow gradually increases in MF-TEG. The reason is that for SSG-VNG, storage links are not exploited, while for MF-TEG, given a larger storage capacity, more data can be stored to wait for suitable opportunities to transmit to the next function node or the destination node, resulting in the increasing of the network maximum flow. Furthermore, we can see that with the increasing storage capacity, the network maximum flow reaches an upper bound in MF-TEG. The reason is that in this case the network communication and computation resources in MF-TEG become the bottleneck and the network performance is dominated by communication and computation resources.

Next, we investigate the impact of the computation capacity of function nodes on the network maximum flow. In Fig. 10, the network maximum flow versus the computation capacity of function nodes is plotted. It can be observed that the network maximum flow in MF-TEG is significantly higher than that in SSG-VNG. The network maximum flow of SSG-VNG slowly increases with the increasing of computation capacity. The reason is that storage links are not exploited in SSG-VNG, so that computation and communication capability cannot match the storage capacity. Furthermore, it can be observed that the network maximum flow in MF-TEG significantly increases with the increasing of computation capacity. The reason is that computation, communication, and storage capability are jointly exploited. Specifically, when the computation capacity is 200 units, the network maximum flow can reach more than 350 Gbit in MF-TEG, while the network maximum flow can only reach less than 75 Gbit in SSG-VNG, which means that in this case, the network maximum flow in MF-TEG is more than 4 times that of SSG-VNG. Furthermore, we can see that with the increasing computation capacity, the network maximum flow gradually increases in both MF-TEG and SSG-VNG. The reason is that given a larger computation capacity, more data can be processed. It can

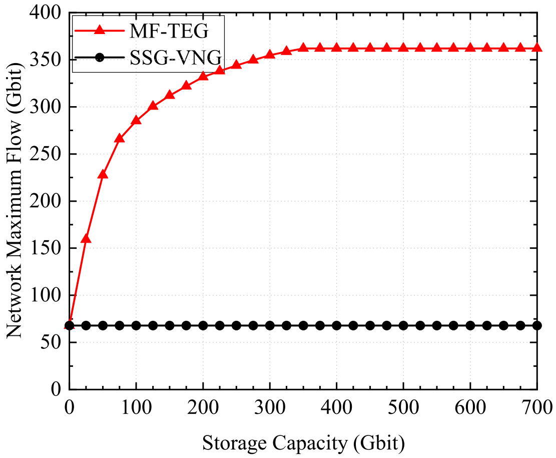


Fig. 9. The performance of the network maximum flow versus the storage capacity.

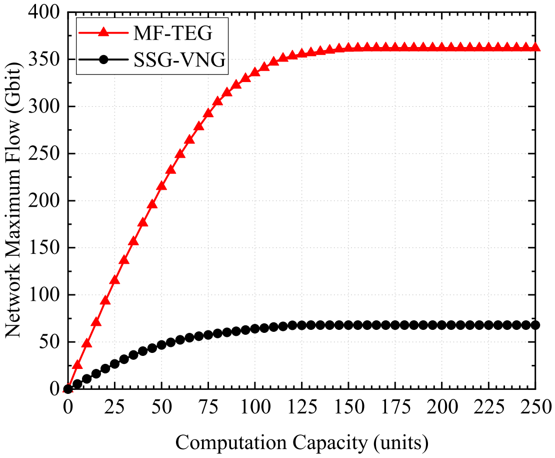


Fig. 10. The performance of the network maximum flow versus the computation capacity.

be further observed that, with the increasing computation capacity, the network maximum flow reaches an upper bound in both MF-TEG and SSG-VNG. The reason is that for the MF-TEG, the network communication and storage resources in this case become the bottleneck and the network performance is dominated by communication and storage resources, while for the SSG-VNG, the network communication resources in this case become the bottleneck and the network performance is dominated by communication resource.

VII. Conclusion

In this paper, we propose the MF-TEG model for jointly modelling the communication, storage, and computing capability of nodes in the dynamic and predictable network over time. We also present the fundamental constraints on the data flow by taking into communication, storage, and computation. The proposed MF-TEG can find many applications, for example, for SGAIN aided multi-tier computing network, it can model SN with SFC, where satellite nodes may provide communication, storage, and multiple computation functions for one mission flow within one same node. The proposed MF-TEG model has the advantage against both the traditional TEG and SSG-aided VNG.

References

[1] X. Shen, W. Liao, and Q. Yin, "A novel wireless resource management for the 6G-enabled high-density Internet of Things," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 32–39, Feb. 2022.

[2] Z. Wang, Z. Zhou, H. Zhang, G. Zhang, H. Ding, and A. Farouk, "AI-based cloud-edge-device collaboration in 6G space-air-ground integrated power IoT," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 16–23, Feb. 2022.

[3] Y. Yang, "Multi-tier computing networks for intelligent IoT," *Nature Electron.*, vol. 2, no. 1, pp. 4–5, Jan. 2019.

[4] K. Wang, W. Chen, J. Li, Y. Yang, and L. Hanzo, "Joint task offloading and caching for massive MIMO-aided multi-tier computing networks," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1820–1833, Mar. 2022.

[5] E. El Haber, T. M. Nguyen, and C. Assi, "Joint optimization of computational cost and devices energy for task offloading in multi-tier edge-clouds," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3407–3421, May 2019.

[6] J. Liu, K. Xiong, D. W. K. Ng, P. Fan, Z. Zhong, and K. B. Letaief, "Max-min energy balance in wireless-powered hierarchical fog-cloud computing networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7064–7080, Nov. 2020.

[7] A. De Domenico, Y.-F. Liu, and W. Yu, "Optimal virtual network function deployment for 5G network slicing in a hybrid cloud infrastructure," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 7942–7956, Dec. 2020.

[8] C. Wang, L. Liu, C. Jiang, S. Wang, P. Zhang, and S. Shen, "Incorporating distributed DRL into storage resource optimization of space-air-ground integrated wireless communication network," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 434–446, Apr. 2022.

[9] S. Li, Q. Chen, W. Meng, and C. Li, "Civil aircraft assisted space-air-ground integrated networks: An innovative NTN of 5G and beyond," *IEEE Wireless Commun.*, vol. 29, no. 4, pp. 64–71, Aug. 2022.

[10] F. Tang, C. Wen, X. Chen, and N. Kato, "Federated learning for intelligent transmission with space-air-ground integrated network (SAGIN) toward 6G," *IEEE Netw.*, early access, Aug. 1, 2022, doi: 10.1109/MNET.104.2100615.

[11] J. Yu, X. Liu, Y. Gao, and X. Shen, "3D channel tracking for UAV-satellite communications in space-air-ground integrated networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2810–2823, Dec. 2020.

[12] Y. Zou, J. Zhu, T. Wu, H. Guo, and H. Wei, "Cooperative drone communications for space-air-ground integrated networks," *IEEE Netw.*, vol. 35, no. 5, pp. 100–106, Sep. 2021.

[13] N. Cheng et al., "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[14] S. Yu, X. Gong, Q. Shi, X. Wang, and X. Chen, "EC-SAGINs: Edge-computing-enhanced space-air-ground-integrated networks for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5742–5754, Apr. 2022.

[15] H. Liao, Z. Zhou, X. Zhao, and Y. Wang, "Learning-based queue-aware task offloading and resource allocation for space-air-ground-integrated power IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5250–5263, Apr. 2021.

[16] B. Cao et al., "Edge–cloud resource scheduling in space-air-ground-integrated networks for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5765–5772, Apr. 2022.

[17] Y. Wang, W. Feng, J. Wang, and T. Q. S. Quek, "Hybrid satellite-UAV-terrestrial networks for 6G ubiquitous coverage: A maritime communications perspective," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3475–3490, Nov. 2021.

[18] M. Sheng, D. Zhou, R. Liu, Y. Wang, and J. Li, "Resource mobility in space information networks: Opportunities, challenges, and approaches," *IEEE Netw.*, vol. 33, no. 1, pp. 128–135, Jan. 2019.

[19] T. Zhang, J. Li, H. Li, S. Zhang, P. Wang, and H. Shen, "Application of time-varying graph theory over the space information networks," *IEEE Netw.*, vol. 34, no. 2, pp. 179–185, Mar. 2020.

[20] H. Yang, W. Liu, H. Li, and J. Li, "Maximum flow routing strategy for space information network with service function constraints," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 2909–2923, May 2022.

[21] D. Fischer, D. Basin, K. Eckstein, and T. Engel, "Predictable mobile routing for spacecraft networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 6, pp. 1174–1187, Jun. 2013.

[22] L. R. Ford and D. R. Fulkerson, "Constructing maximal dynamic flows from static flows," *Oper. Res.*, vol. 6, no. 3, pp. 419–433, 1958.

[23] L. R. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton, NJ, USA: Princeton Univ. Press, 1962.

[24] W.-K. Chen, Y.-F. Liu, A. De Domenico, Z.-Q. Luo, and Y.-H. Dai, "Optimal network slicing for service-oriented networks with flexible routing and guaranteed E2E latency," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 4, pp. 4337–4352, Dec. 2021.

[25] I. Jang, D. Suh, S. Pack, and G. Dán, "Joint optimization of service function placement and flow distribution for service function chaining," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2532–2541, Nov. 2017.

[26] N. Zhang, Y.-F. Liu, H. Farmanbar, T.-H. Chang, M. Hong, and Z.-Q. Luo, "Network slicing for service-oriented networks under resource constraints," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2512–2521, Nov. 2017.

[27] X. Gao, R. Liu, and A. Kaushik, "Virtual network function placement in satellite edge computing with a potential game approach," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 2, pp. 1243–1259, Jun. 2022.

[28] Y. Li, C. Song, D. Jin, and S. Chen, "A dynamic graph optimization framework for multihop device-to-device communication underlaying cellular networks," *IEEE Wireless Commun.*, vol. 21, no. 5, pp. 52–61, Oct. 2014.

[29] Q. Hao, M. Sheng, D. Zhou, and Y. Shi, "A multi-aspect expanded hypergraph enabled cross-domain resource management in satellite networks," *IEEE Trans. Commun.*, vol. 70, no. 7, pp. 4687–4701, Jul. 2022.

[30] C. Jiang and X. Zhu, "Reinforcement learning based capacity management in multi-layer satellite networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4685–4699, Jul. 2020.

[31] Z. Jia, M. Sheng, J. Li, D. Zhou, and Z. Han, "VNF-based service provision in software defined LEO satellite networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 6139–6153, Sep. 2021.

[32] B. Du, X. Di, D. Liu, and H. Zhang, "Dynamic graph optimization and performance evaluation for delay-tolerant aeronautical ad hoc network," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6018–6036, Sep. 2021.

[33] D. Zhou, M. Sheng, X. Wang, C. Xu, R. Liu, and J. Li, "Mission aware contact plan design in resource-limited small satellite networks," *IEEE Trans. Commu.*, vol. 65, no. 6, pp. 2451–2466, Mar. 2017.

[34] Y. Wang et al., "Multi-resource coordinate scheduling for earth observation in space information networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 268–279, Feb. 2018.

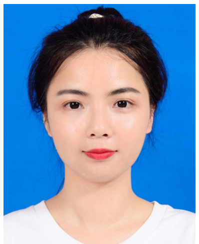
[35] Y. F. Liu, Y. H. Dai, and Z. Q. Luo, "Joint power and admission control via linear programming deflation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1327–1338, Mar. 2013.

[36] Gurobi Optimization. *Gurobi Optimizer Reference Manual*. Accessed: Jan. 6, 2023. [Online]. Available: http://www.gurobi.com

[37] Iridium Communications Inc. *Iridium Next: A Global Effort to Launch the Future of Global Communications*. Accessed: Jan. 6, 2023. [Online]. Available: https://www.iridium.com/network/

[38] LeoSat Enterprises. *A New Type of Satellite Constellation*. Accessed: Jan. 6, 2023. [Online]. Available: https://www.leosat.com/to/technology/

Wei Liu (Senior Member, IEEE) received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 1999, the M.S. degree from Xidian University, Xi'an, China, in 2003, and the Ph.D. degree from the University of Southampton, U.K. Since 2007, he has been with Xidian University. His current research interests include MIMO communications systems, interference management, joint communication, and computation and caching.



Huiting Yang received the B.E. degree in communications engineering and the M.S. degree in space science and technology from Xidian University, Xi'an, China, in 2015 and 2018, respectively, where she is currently pursuing the Ph.D. degree. Her current research interests include resource allocation for space information networks, space-air-ground integrated networks, and vehicular networks.

Jiandong Li (Fellow, IEEE) received the M.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 1985 and 1991, respectively. Since 1985, he has been a Faculty Member with the School of Telecommunications Engineering, Xidian University, where he is currently a Professor. From 2002 to 2003, he was a Visiting Professor with the Department of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA. His main research interests include wireless communication theory, cognitive radio, and signal processing. He is a fellow of the China Institute of Electronics and the China Institute of Communication. From 1993 to 1994, and from 1999 to 2000, he was a member of Personal Communications Networks, Specialist Group for China 863 Communication High Technology Program. He is also a member of Specialist Group of the New Generation of Broadband Wireless Mobile Communication Networks for the Ministry of Industry and Information Technology, and the Chair of Broadband Wireless IP Standard Work Group, China. He was awarded as a Distinguished Young Researcher from NSFC and a Changjiang Scholar from the Ministry of Education, China. He was the General Vice Chair of ChinaCom 2009 and the TPC Chair of IEEE ICCC 2013.