Satellite Mobile Edge Computing: Improving QoS of High-Speed Satellite-Terrestrial Networks Using Edge Computing Techniques

Zhenjiang Zhang, Wenyu Zhang, and Fan-Hsun Tseng

Abstract

The high-speed satellite-terrestrial network (STN) is an indispensable alternative in future mobile communication systems. In this article, we first introduce the architecture and application scenarios of STNs, and then investigate possible ways to implement mobile edge computing (MEC) technique for QoS improvement in STNs. We propose satellite MEC (SMEC), in which a user equipment without a proximal MEC server can also enjoy MEC services via satellite links. We propose a dynamic network virtualization technique to integrate the network resources, and furtherly design a cooperative computation offloading (CCO) model to achieve parallel computation in STNs. Task scheduling models in SMEC are discussed in detail, and an elemental simulation is conducted to evaluate the performance of the proposed CCO model in SMEC.

Introduction

Although 4G LTE cellular mobile networks (MONETs) have achieved great advances in providing fast and convenient communication services for mobile terminals, over 50 percent of the world, especially rural areas and isolated islands, still lack high-speed Internet access services [1]. Moreover, MONET facilities are vulnerable to natural disasters, such as earthquakes and hurricanes. To provide ubiquitous, broadband, and reliable Internet connections, in recent years satellite-terrestrial networks (STNs) have attracted great interest in both academic and practical circles [2].

In recent years, satellite technologies have achieved great improvements in terms of manufacturing, spot-beam antenna, and laser transmission, which make satellites, especially low Earth orbit (LEO) satellites, much more economical, miniaturized, and high-throughput. In addition, the advancement of antenna techniques enables the satellite to fully exploit Ku and Ka bands to broaden the spectrum pools. Currently, the latency from a terrestrial station to its visible LEO satellites can be reduced to 1–4 ms. These low-cost, low-latency, and high-throughput LEO satellites have greatly promoted the development of STNs [3], and several ambitious STN projects have been launched, such as OneWeb [1], SpaceX Starlink [4], and O3b [5]. For example, the OneWeb project plans to build a constellation with over 600 LEO satellites to provide global high-speed Internet access services by 2027, and it is planned to be compatible with existing 3G/4G/WiFi mainstream network access services. With the support of these high-speed STNs, it is foreseeable that broadband wireless access to the Internet can be truly ubiquitous in the near future.

Similar to MONEts, STNs are also facing the challenges of users' increasing demands for higher quality of services (QoS), such as high data rates, low communication latency, and low energy cost on data communication and processing. The evolution of smartphones and tablets has hastened many new computation and energy intensive applications, such as speech recognition, gaming, multimedia encoding/decoding, and intelligent transportations. Although cloud computing can be used to fulfill these tasks, it suffers from the delay or delay-jitter of the unreliable data transmission in the wide area network (WAN) [6]. As such, mobile edge computing (MEC) has been proposed to transfer the computation and storage capacity from cloud computing to local servers, and is able to greatly improve the QoS to users, such as reducing user-perceived latency and energy consumption of user devices [7]. In addition, MEC can provide content caching and storage services, which are effective in reducing the traffic amount of STNs.

So far, existing works have mainly focused on investigating MEC in conventional cellular MONEts that directly connect to the Internet through the gateway of a radio access network (RAN). As an important alternative to support mobile communication, high-speed STNs can also benefit from the advantages of MEC, and we call this edge computing technique for STNs satellite mobile edge computing (SMEC). Although some advanced techniques, such as software defined networking (SDN) [8], network functions virtualization (NFV) [9], cognitive radio [10], and content caching [11], have been proposed to improve the performance of STNs, there is still a lack of a comprehensive discussion on using MEC to improve the QoS of STNs, especially from the perspective of computation offloading.

The scenario of implementing MEC in STN is quite different from conventional MONETs. In STNs, unlike densely distributed users, sparsely distributed mobile users may connect to the Internet via a small terrestrial terminal station, which has no extra computation and storage capacities, and deploying a domain SMEC server for a small

Digital Object Identifier:
10.1109/MNET.2018.1800172

Zhenjiang Zhang and Wenyu Zhang are with Beijing Jiaotong University; Fan-Hsun Tseng is with National Taiwan Normal University.

0890-8044/19/$25.00 © 2019 IEEE

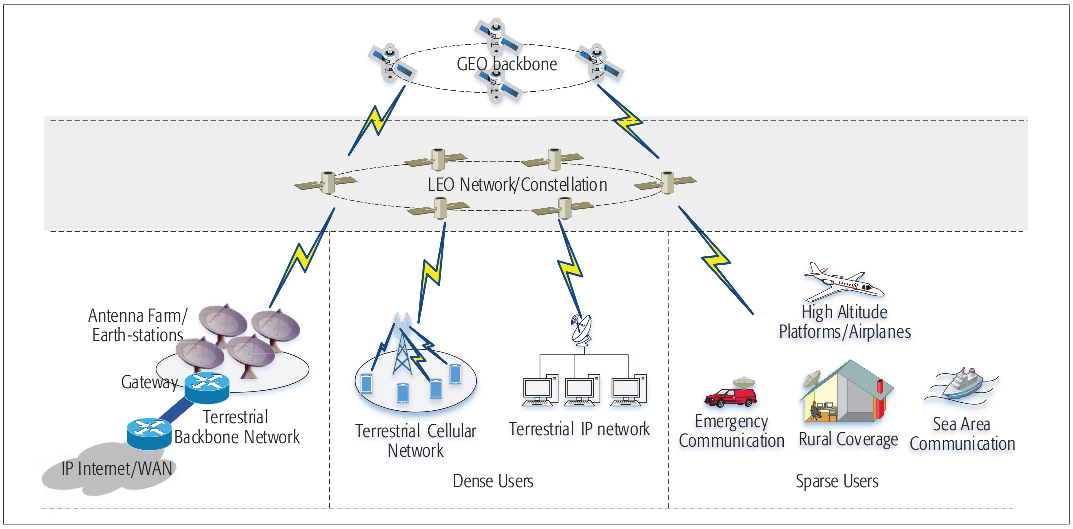

FIGURE 1. Architecture of a high-speed STN: its application scenarios.

number of users is economically unrealistic. Moreover, the coverage area of a LEO satellite is much larger than the regional RANs in MONETs, and how to provide MEC services for UEs of a wide coverage in STNs needs to be carefully considered.

In this article, we propose to implement MEC techniques in high-speed STNs to improve the QoS of the STN mobile users. We first introduce the architecture and application scenarios of STNs, and discuss possible services of SMEC. Particularly, we focus on computation offloading in SMEC, and investigate possible ways to implement SMEC in STNs. We propose to use a dynamic NFV technique to integrate the resources of the network, and then propose a cooperative computation offloading (CCO) model specifically for STN-based mobile communication systems. Possible task scheduling models are discussed, and an elemental simulation is provided to demonstrate the performance of the CCO model.

High-Speed STNs

Figure 1 shows the basic architecture of an STN and its application scenarios. The detailed network infrastructure and application scenarios are illustrated as follows.

Network Infrastructure

LEO Constellation: Unlike the eNodeB in mobile cellular systems, the energy supply and computation capacity of a LEO satellite are quite limited; thus, tasks with high energy consumption or intensive computations may not be practical in small-scale, especially micro-scale, LEO satellites. Note that the LEO satellite is not the only option in high-speed STNs; for example, the o3b network [5] operates a medium Earth orbit (MEO) satellite constellation to provide relay services for data transmission.

GEO Backbone Network: For a space-terrestrial integrated network (STIN), the GEO backbone network plays an important role in relaying data for LEO or MEO satellite networks. However, with the development of laser communication technique for inter-satellite communications, the GEO relay satellite is not a necessary part of an STN. Moreover, relaying by GEO satellites may cause much higher delay compared to inter-satellite communications.

High-speed STNs can be extended to build neural networks for Internet or mobile communication service providers. When the users are densely distributed in a region, connections to the satellite can be achieved by using terrestrial stations with more powerful antennas.

Terrestrial Station (TS): In an STN, there are the following two types of TSs: the terrestrial gateway stations (TGSs) and terrestrial terminal stations (TTSs). A TGS is supposed to be equipped with powerful directional antennas, and it is able to support a large group of users, such as a cellular network or a regional IP network. On the contrary, the antenna of a TTS is not that powerful, and it is just used to relay data for a small group of users, such as a family or a school.

Terrestrial Backbone Network (TBN): The total traffic of a wide terrestrial area of an STN is aggregated in TBNs. To improve the throughput, an antenna farm or several Earth stations can be deployed. The traffic between the IP Internet and the TBN is transported through the Internet gateways, and traffic between the TBN and satellites is transported through the Internet-to-orbit gateways. Note that an STN may have several TBNs on the ground. In this article, a TBN is regarded as having sufficient computing and storage capacities and energy supply to support the offloading services for the users.

Application Scenairos

Dense Users: High-speed STNs can be extended to build neural networks for Internet or mobile communication service providers. When users are densely distributed in a region, connections to the satellite can be achieved by using terrestrial stations with more powerful antennas. For mobile users, 4G/5G services can be provided by deploying cellular towers and eNBs to construct terrestrial cellular networks. An IP regional area network (IPRAN) can be constructed by deploying gateways that are connected to terrestrial stations. The Internet-orbit gateways can be used to build stable links between terrestrial networks and satellites. In this way, users without powerful antennas

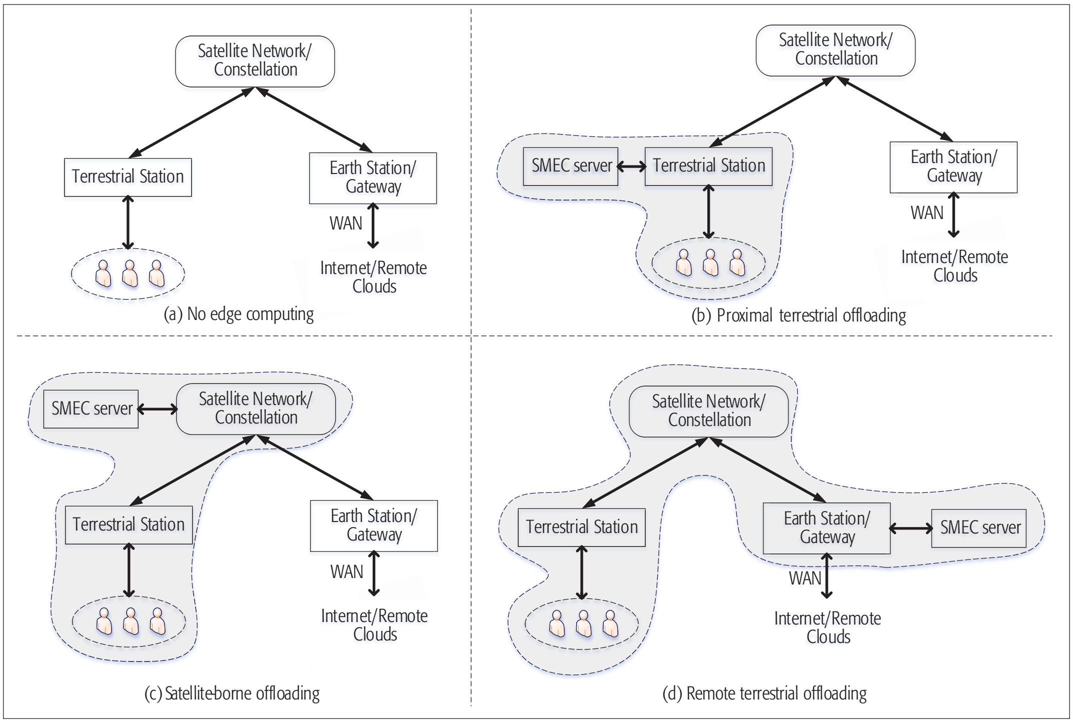

FIGURE 2. Possible ways of computation offloading in SMEC

can enjoy high-speed Internet connection through the relaying of terrestrial stations.

Sparse Users: When users are sparsely distributed, connections with the LEOs can be achieved through their own capacity-limited TTSs. Possible users include high altitude platforms and airplanes, emergency communication, and rural and sea area coverage. In these cases, the communication, computation, caching, and storage capacities of the terminals are quite limited. As such, resource limited terrestrial terminals are not suitable for executing the offloading tasks.

Satellite Mobile Edge Computing

In this section we introduce the benefits of SMEC in STNs, and investigate possible ways to implement computation offloading functions under the framework of STNs.

What Can SMEC Do in STNs?

Similar to MEC in cellular MONETs, SMEC has the following two main advantages: one is computation offloading, which is able provide extensive computation capacity for user devices; the other is content caching/storage, which is able to cache contents and store data files to reduce repeat transmissions from remote Clouds.

Computation Offloading: Due to the bottleneck transmission of A WAN, delay or delay-jitter is unavoidable in cloud computing. For resource-limited mobile STN users, offloading computation tasks to some local MEC servers will greatly reduce the delay, since interactions with remote clouds are not needed. For some delay-sensitive applications, such as gaming and vehicle networks, user-perceived delay will be significantly reduced. In addition, for some computation-costly applications, the computation capacity is greatly enhanced for mobile user devices, such as smartphones and tablets, and energy consumption is also reduced.

Instead of local execution, there are the following two computation offloading modes [12]. The first one is full offloading, in which all the intensive computation tasks are offloaded onto the MEC server. This mode is suitable when the battery of a user device is not sufficient and energy consumption for data transmission is not high. The second one is partial offloading, in which a user decides whether to offload the task to an MEC server. A simple policy for partial offloading is that if the delay and data transmitting energy of the data file are acceptable, the task is executed in the MEC server; otherwise, in user devices.

Content Caching/Storage: The explosive growth of user data, especially multimedia data, has brought considerable stress on mobile networks [13]. The challenge of traffic explosion is also severe in high-speed STNs. In addition to improving the bandwidth of RANs in MONETs, using caching to reduce data transmission is a very effective means, and over half of HTTP contents are cacheable. By caching the contents in satellites or terrestrial stations, retransmissions of the same contents can be avoided, and the traffic of STNs will be significantly reduced. In addition, content caching can be used to speed up html web pages and other applications, and further reduce the user-perceived delay. Besides web-based caching, it is also useful for providing extra storage capacity to support the storage of audio or video multimedia data files. If contents are pre-cached or pre-downloaded in SMEC servers, redundancy elimination (RE) can be used to reduce unnecessary traffic.

Where to Offload?

In STNs, users access the Internet by the relaying of both terrestrial stations and LEO satellites. Providing computation offloading services for the users in such a wide area is different from MON-

ETs. As shown in Fig. 2, computation offloading in STNs may have the following possibilities.

Proximal Terrestrial Offloading (PTO): SMEC servers are deployed in terrestrial stations that are proximal to the users, as shown in Figure 2b. This mode is very similar to conventional MEC. If computation tasks are offloaded to SMEC servers, the latency can be significantly reduced since the backhaul transmission to the satellite is avoided. Besides, the traffic of the whole STN will also be relieved. However, this PTO mode requires that the terrestrial stations are equipped with extensive computation capacities, which is suitable for terrestrial station-relayed communication for dense users. For sparse users that connect to satellites via micro terrestrial terminal stations (TTSs), PTO may not be practical, since the TTSs have no extra computation capacities. Moreover, deploying such a dedicated MEC server for a small group of users is not economically viable.

Satellite-Borne Offloading (SBO): in which SMEC servers are deployed in LEO satellites, as shown in Fig. 2c. In this situation, all the member terrestrial users of the satellite can benefit from offloading services. The traffic between satellite and terrestrial backbone network (TBN) is also reduced. Compared to PTO, the latency of this scheme is relatively higher, but it is still a significant improvement in latency compared to sending requests to remote clouds. The drawback of this scheme is that the embedding of such an SMEC server in a satellite will increase its hardware cost. Most importantly, conducting intensive computations will significantly increase the energy consumption of the satellite, which may not be practical when energy supply of the satellites is strictly limited.

Remote Terrestrial Offloading (RTO): SMEC servers are deployed in energy-sufficient TBN gateways that connect to the IP Internet, as shown in Fig. 2d. If an SMEC server is deployed in the TBN gateway, all the domain users can enjoy SMEC services, and data transmission between TBN and remote clouds through a WAN is avoided. Note that compared to an eNB, the SMEC server should be more powerful, or the heavy workload may not be able to be completed. Compared to PTO and SBO, the delay of RTO is relatively higher due to the extra relaying of LEO satellite links, but it is more practical for implementation and maintenance.

Dynamic NFV

In a cellular MONET, a radio access network (RAN) is usually assigned with one independent MEC server. Even when there are several MECs in a RAN, they can be virtually unified as one by exploiting NFV. Different from MONETs, STN-based cellular networks use LEO satellites to build links among users and WAN. Exploiting the NVF technique to unify available resource enables the network to centrally manage the computation offloading tasks. Besides, NFV provides great advantages in reducing the cost and time of deploying new services in SMECs for the network providers [14]. However, in STNs, all the LEO satellites are in high-speed motion. For a terrestrial station, the visible satellites are varying in different time intervals. In this situation, the member SMEC servers in the coverage of a satellite are always changing; thus, it is hard or even impractical to implement static NFV to unify the SMEC servers.

As such, we propose to use a dynamic NFV technique to integrate the resources of STNs, as shown in Fig. 3. The dynamic virtualization system is similar to the commonly used static NFV systems, containing three layers: the virtualized infrastructure layer, the virtualized network function (VNF) layer, and the orchestrator layer. To deal with mobility of LEO satellites, a dynamic resource monitor is added in the satellite orches-

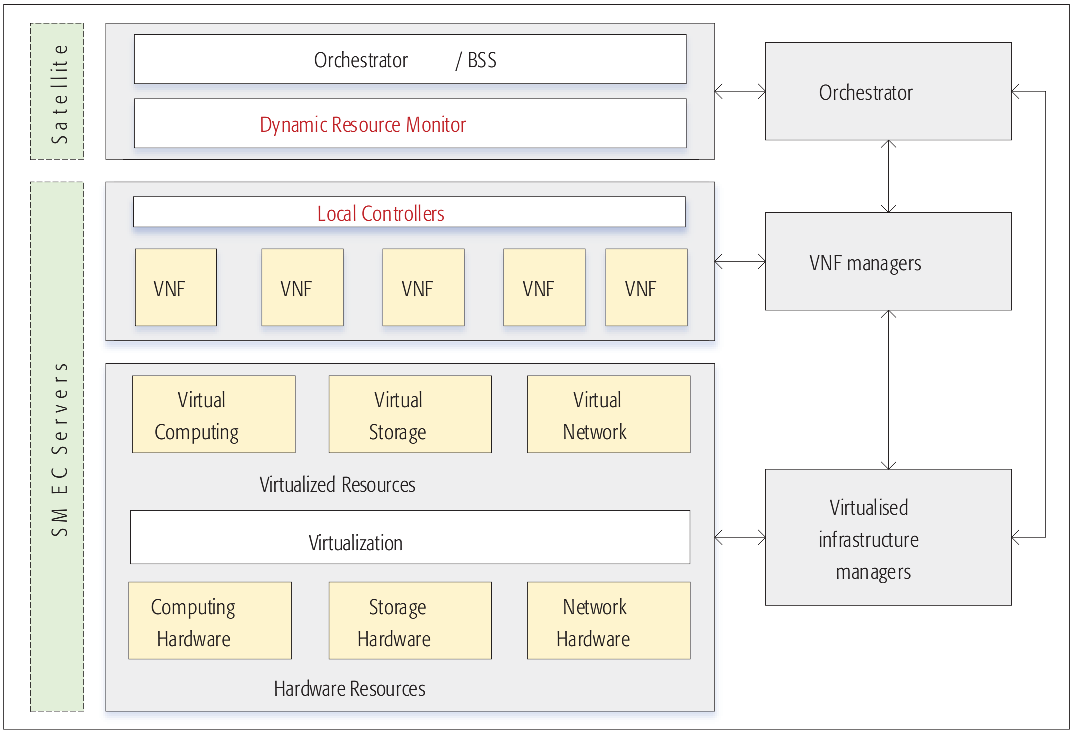

FIGURE 3. Dynamic NFV architecture in SMEC.

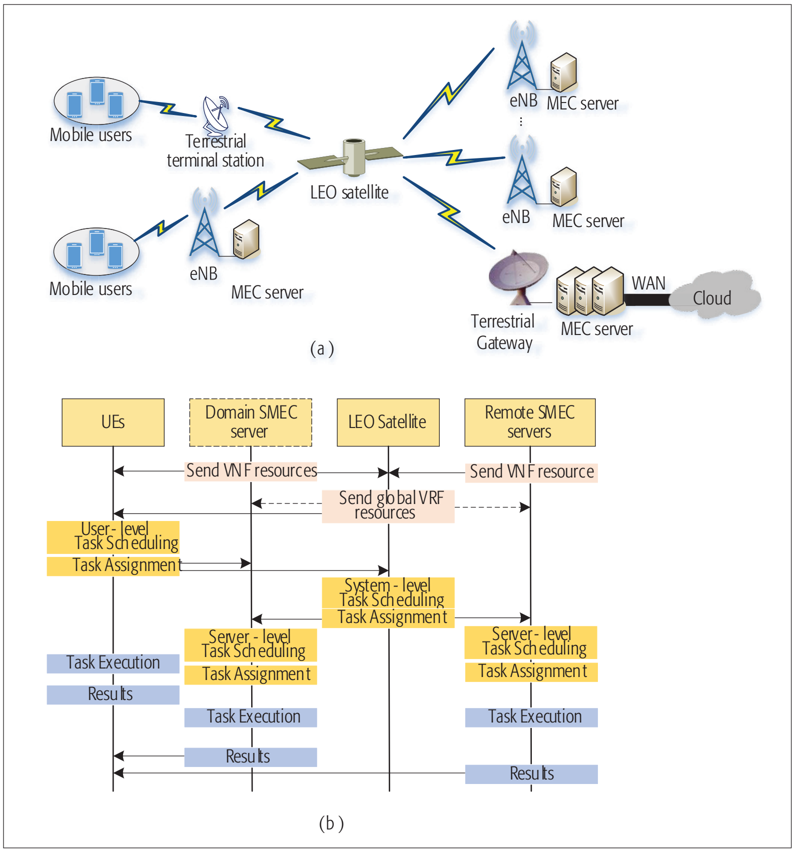

FIGURE 4. Cooperative computation offloading in SMEC: a) the system architecture; b) flow diagram of the CCO in STN.

trator. In this way, the monitor knows whether a VNF is connected or gone, and resource registration and deletion operations can be quickly done in the orchestrator. On the other hand, when the available resources have changed, the information will be sent to the users and MEC servers to adjust their policies.

Cooperative Offloading

Based on the dynamic NFV-based resource integration scheme, we further propose cooperative computation offloading (CCO), in which the SMEC servers cooperates with each other to complete computation tasks of the user tasks. An illustrative example of CCO in STNs is shown in Fig. 4, in which Fig. 4a depicts the system architecture, and Fig. 4b plots the flow diagram of the CCO model. In Fig. 4a, the users can offload the tasks to its domain SMEC server or other remote SMEC servers in the coverage area via satellite links. In Fig. 4b, task offloading in CCO mainly includes the following phases.

Resource Integration: The user first receives the virtualized computation resource information from its proximal SMEC server. At the same time, the LEO satellite integrates the VNF resources, and sends all the available virtualized computing resources to the users and SMEC servers in its coverage area. Note that the SMEC servers may also need to know the global VNF resources if they want to migrate tasks to other SMEC servers.

Task Scheduling and Assignment: Task scheduling can be conducted to distribute pending tasks to the available SMECS. User-level, server-level, and system-level task scheduling can be conducted in user terminals, SMEC servers, and LEO satellites, respectively. Note that the above three-level task scheduling operations can be conducted independently and simultaneously to optimize their own performance.

Task Execution: Obviously, if the proximal SMEC server has sufficient computation capacity, the tasks will be more probable to be assigned to the proximal SMEC, since it has lower latency compared to other remote SMECs. At last, the obtained results are returned to the user terminals.

Task Scheduling in SMEC

In this section, we discuss potential models for task scheduling problems in SMEC. Exemplary simulation results will also be provided demonstrate the performance of SMEC.

Scenarios

According to the execution location, task scheduling in SMEC can be divided into the following categories:

User-Level Task Scheduling: The users decide whether to offload the pending tasks to SMEC servers themselves. Real-time information of available computing resources, including the user device and the computing virtual machines (VMs) of the SMECs, must be known to the users. The related information can be energy cost and time in data transmission and task executions. Note that wireless channel state and radio access ways can also be considered in the optimization model.

Server-Level Task Scheduling: An SMEC server can conduct task scheduling operations to maximize its performance on energy consumption and user-perceived delay. This optimization is suitable for servers with multiple computing VMs. The system state information, including buffer size, execution time, and energy cost information, need to be dynamically known to the adaptive load dispatcher of the server.

System-Level Task Scheduling: This can be realized by enabling the satellite, or a proxy SMEC server, to assign the tasks to its member SMECs in the coverage area. Objectives can be minimizing expected task execution delays or system-level energy cost, or maximizing the system-level energy efficiency. Note that server-level and system-level task scheduling can also be conducted along with the resource management operations, such as adjusting the processing rates and spectrum utilization efficiency.

Objectives

To maximize the performance of the SMEC system, task scheduling can be conducted to assign proper destinations for pending tasks. In SMEC, the time and energy costs of the inter-satellite link and space-terrestrial links also need to be considered, which leads to a more complex scheduling model compared to MEC. According to objectives, the management model can be categorized as follows.

Minimization of User-Perceived Delay: The objective can be to find a proper assignment strategy to minimize the expected user-perceived delays. However, the energy constraint needs to be properly specified, since some tasks with high energy transmission cost are not suitable to be offloaded to SMECs.

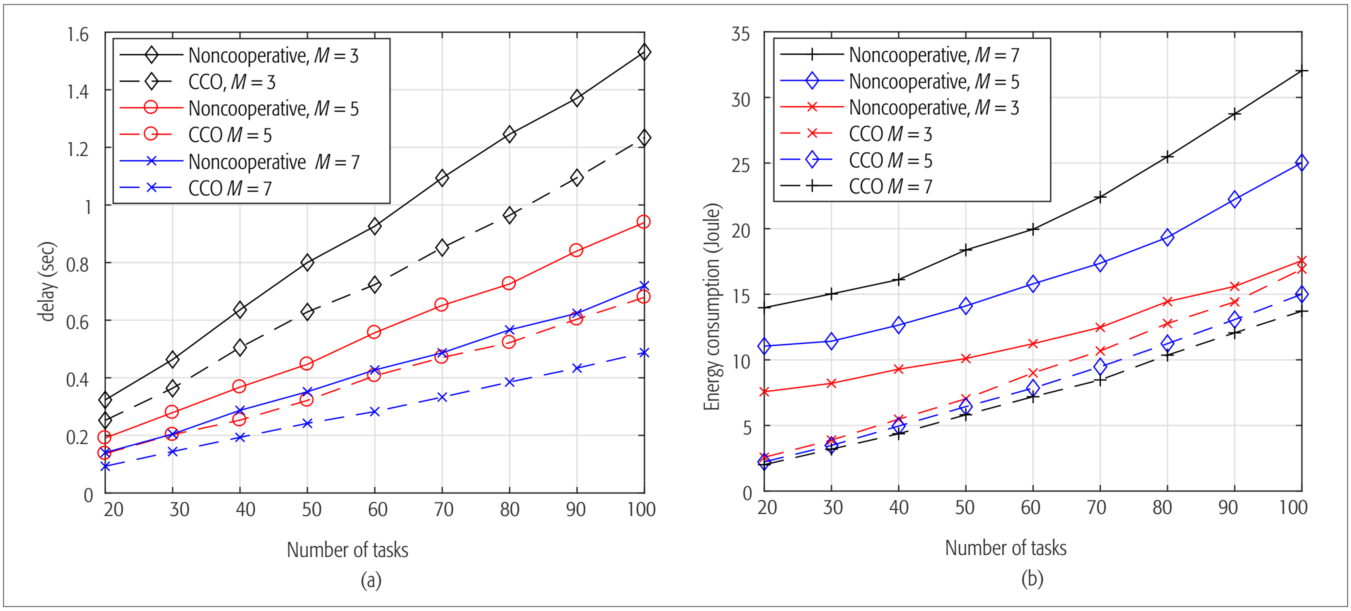

FIGURE 5. Performance evaluation of CCO in SMEC: a) user-perceived delay; b) energy consumption of the system.

Minimization of Energy Consumption: The objective is to find an optimal assignment strategy to minimize the expected energy consumption. Compared to terrestrial communication, the energy limitation is much more severe in satellite. In this scenario, the delay constraint needs to be properly specified, since some tasks with high execution or transmission time are not suitable to be offloaded to SMEC servers.

Maximization of Energy Efficiency: The trade-off between the energy consumption and user-perceived delay has always existed in edge computing systems. It is meaningful to find a proper task allocation scheme that maximizes energy efficiency. In this way, the system can maximize the amount of executed tasks with a given amount of energy supply.

Constraints

The possible constraints in SMEC task scheduling are summarized as follows.

Energy: This constraint ensures that the energy cost of computing offloading is not larger than a predefined threshold. For example, if a data file size of a task is too large, it may not be reasonable to be offloaded to SMEC servers, since transmission will cost a large amount of energy. Energy cost in SMEC is more complex to model compared to MEC. Energy cost may be caused by terrestrial transmission, space-terrestrial transmission, and inter-satellite communication. In particular, since the satellites are usually power-limited, it is crucial to save energy cost as much as possible.

Delay: For a computation-intensive task, it is crucial for users to consider the delays caused by compression, transmission, and task execution operations. The tolerable delay can be predefined according to the actual demand of the applications. It is important to note that the delays in STN may be larger compared to a RAN in a MONET, since the throughput of a LEO satellite is usually not much larger than that of eNBs. Therefore, compared to MEC, the delay constraint should be relatively looser.

Channel/Time Slot Occupation: In wireless communications, proper usage of the available channels and time slots needs to be carefully considered. In a scenario with multiple users, a channel is able to support at most one user within a time slot, and the maximum user device number should not be larger than the total channel number. However, the bandwidth of satellite communication is usually much more wider than cellular networks, and orthogonal frequency-division multiplexing (OFDM) is not usually used. Therefore, channel allocation may not need to be considered.

User Priority: In practice, the tasks are assigned different priorities to satisfy the quality of experience (QoE) of different users. Task differentiation can be naturally considered in the CCO framework: a task with a high-level priority is more likely to be processed in its domain MEC service, while a task with lower priority is more likely to be migrated to other remote SMEC servers. To achieve this goal, each task will be assigned a weight to represent its priority. When a task is more important, the weight will be larger, and vice versa.

Performance Evaluation

We consider an SMEC system with multiple computing servers. By exploiting the NFV techniques, the computing resources are virtualized to different M independent VMs. Suppose that there are N pending tasks to be processed, and task file sizes and CPU cycle demands of the tasks are denoted as $b_1, \ldots, b_N$ and $d_1, \ldots, d_N$, respectively. When the objective is to minimize the expected user perceive delay, we have

$$\min \sum_{i=1}^{N}\sum_{j=1}^{M} u_{ij}(2b_{ij}\mathbb{E}(t_{ij}^{tr})+d_i\mathbb{E}(t_{ij}^{cp}))$$

$$s.t. \sum_{j=1}^{M} u_{ij}=1, \forall i=1,\ldots,N$$

where t_{ij}^{tr} denotes the unit transmission time of task i to MEC j, and t_{ij}^{cp} is the unit computation time of task i in VM j. When the task is processed locally in a user device, there is $t_{ij}^{tr}=0$. Note that the queening time is also included in t_{ij}^{tr}. $\mathbb{E}(x)$

> It can be seen that when given a fixed number of SMEC servers, the performance improvements is more evident when the number of UEs get larger. Therefore, it is meaningful to integrate the resources in STNs to reduce user-perceive delays and energy consumption of the SMEC servers.

denotes the expectation of random variable x. The matrix $\mathbf{u} = [u_{ij}]_{M \times M}$ contains the task offloading decisions (i.e., u_{ij} is the decision whether task i is assigned to VM j). When task i is assigned to VM j, we have $u_{ij} = 1$; otherwise, $u_{ij} = 0$. Specifically, $u_{i1} = 1$ means that the task will be executed locally in user devices. The equation constraint ensures that each task is assigned with one and only one computation VM.

In a similar way, the objective can also be minimizing the total energy consumption, which is

$$e_{total} = \sum_{i=1}^{N}\sum_{j=1}^{M}(2b_i\mathbb{E}(e_{ij}^{tr}) + d_i\mathbb{E}(e_{ij}^{cp})),$$

where e_{ij}^{tr} denotes the unit transmission energy consumption of task i to j, and e_{ij}^{cp} is the unit computation energy consumption of task i spent in MEC. In this scenario, an inequality delay constraint needs to be added to ensure the execution delay is lower than a predefined threshold.

We conduct an exemplary simulation to evaluate the performance of CCO in SMEC. We assume that there are multiple terrestrial stations, and each of them has an MEC server. The results are shown in Fig. 5, in which M is the number of computation VMs, and Figs. 5a and 5b plot the user-perceived delay and system energy consumption with increasing number of tasks, respectively. It is apparent that using an optimal scheduling model can significantly reduce the delays and energy consumption caused by task offloading. More specifically, it can be seen that when given a fixed number of SMEC servers, the performance improvements are more evident when the number of UEs gets larger. Therefore, it is meaningful to integrate the resources in STNs to reduce user-perceived delays and energy consumption of the SMEC servers.

Conclusions

In this article, we have proposed to integrate the SMEC technique in STN-based mobile communication systems, and investigated implementation methods of SMEC. We have proposed a dynamic NFV technique to integrate the computing resources in the coverage of a LEO satellite, and further proposed a cooperative computation offloading method specifically for the SMEC scenarios. Possible task scheduling models have been discussed, and simulation results demonstrate that the proposed model is able to significantly reduce the user-perceived delay and system energy consumption.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grant 61772064, the Fundamental Research Funds for the Central Universities under grant 2017YJS026, and the Ministry of Science and Technology (MOST), R.O.C. under Grant no. MOST107-2636-E-003-001.

References

[1] T. Azzarelli, Oneweb Global Access//ITU Int'l. Satellite Symp., 2016.

[2] J. P. Cho and C. Joo, "Challenges for Efficient and Seamless Space-Terrestrial Heterogeneous Networks," *IEEE Commun. Mag.*, vol. 53, no. 5, May 2015, pp. 156–62.

[3] M. Gregory *et al.*, "TESAT Laser Communication Terminal Performance Results on 5.6 Gbit Coherent Inter Satellite and Satellite to Ground Links," *Int'l. Conf. Space Optics 2010*, Nov. 2017, vol. 10565, p. 105651F; Int'l. Soc. Optics and Photonics.

[4] V. L. Foreman, A. Siddiqi, and O. D. Weck, "Large Satellite Constellation Orbital Debris Impacts: Case Studies of OneWeb and SpaceX Proposals," AIAA Space and Astronautics Forum and Expo, 2017.

[5] L. Wood, Y. Lou, and O. Olusola, "Revisiting Elliptical Satellite Orbits to Enhance the O3b Constellation," *Physics*, vol. 67, 2014, pp. 110–18.

[6] W. Zhang, Z. Zhang, and H. C. Chao, "Cooperative Fog Computing for Dealing with Big Data in the Internet of Vehicles: Architecture and Hierarchical Resource Management," *IEEE Commun. Mag.*, vol. 55, no. 12, Dec. 2017, pp. 60–67.

[7] Y. Mao *et al.*, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Commun. Surveys & Tutorials*, 2017.

[8] R. Ferrús *et al.*, "SDN/NFV-Enabled Satellite Communications Networks: Opportunities, Scenarios and Challenges," *Physical Commun.*, vol. 18, no. P2, 2016, pp. 95–112.

[9] M. Sheng *et al.*, "Toward a Flexible and Reconfigurable Broadband Satellite Network: Resource Management Architecture and Strategies," *IEEE Wireless Commun.*, vol. 24, no. 4, Aug. 2017, pp. 127–33.

[10] K. Liolis *et al.*, "Cognitive Radio Scenarios for Satellite Communications: The CoRaSat Approach," *IEEE Future Network and Mobile Summit*, 2013, pp. 1–10.

[11] L. Galluccio, G. Morabito, and S. Palazzo, "Caching in Information-Centric Satellite Networks," vol. 11, no. 18, 2012, pp. 3306–10.

[12] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Commun. Surveys & Tutorials*, 2017, vol. 99, pp. 1-1.

[13] S. Wang *et al.*, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," *IEEE Access*, vol. 5, no. 99, 2017, pp. 6757–79.

[14] R. Mijumbi *et al.*, "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE Commun. Surveys & Tutorials*, 2017, vol. 18, no. 1, pp. 236–62.

Biographies

Zhenjiang Zhang received his Ph.D. degree in communication and information systems from Beijing Jiaotong University (BJTU), China, in 2008. He has been a professor in the same university since 2014. He currently serves as the vice dean of the School of Software Engineering in BJTU. He has published about 70 professional research papers. His research interests include fog computing, communication protocols, and wireless sensor networks.

Wenyu Zhang received his B.S. and M.S. degrees in communication engineering from BJTU in 2013 and 2016, respectively, where he is currently working toward a Ph.D. degree in communication engineering. His research interests include protocols and resource management in edge computing and vehicle networks.

Fan-Hsun Tseng [S'12, M'18] received his Ph.D. degree in computer science and information engineering from National Central University, Taiwan, in 2016. He is currently an assistant professor in the Department of Technology Application and Human Resource Development, National Taiwan Normal University, Taipei. His research interests include mobile edge computing, 5G mobile networks, and artificial intelligence.