frontiers in Communications and Networks

PERSPECTIVE
published: 14 September 2021
doi: 10.3389/frcmn.2021.717476

A Software-Defined Architecture for Integrating Heterogeneous Space and Ground Networks

Jie Sun[1], *Feng Liu*[1]*, *Yong Li*[2], *Lianlian Zhang*[1] *and Dingyuan Shi*[1]

[1]School of Electronics and Information Engineering, Beihang University, Beijing, China, [2]Department of Electronic Engineering, Tsinghua University, Beijing, China

In recent years, various types of heterogeneous networks develop rapidly. The integration of multi-type networks have great values in the fields of military and civil applications. The challenges of integrating multiple networks covers the heterogeneity of multiple aspects, e.g., the architectures, protocols, and switching mechanisms. The existing interconnection technologies of heterogeneous networks mainly include traditional static protocol gateways, traditional software-defined network (SDN) gateways, and improved SDN gateways. However, traditional static protocol gateways need to be customed in advance according to specific scenarios, which leads to the lack of flexibility. Traditional SDN gateways are often used for connecting homogeneous networks. The existing improved SDN gateways often neglect the efficiency and cost of integrating heterogeneous networks. In our work, we propose a software-defined architecture for integrating heterogeneous space and ground networks (SD-SGN). First, we propose an integrated architecture that utilizes SDN gateways and southbound interfaces to shield subnets' heterogeneity ranging from the physical layer to the network layer. Second, we use the multi-class multi-level flow tables to provide a flexible data plane. Third, we offer an efficient control plane based on the subnet abstraction and global collaborative optimization. Fourth, we give a further discussion on customizing a complete network service based on the proposed SDN architecture. Last, extensive simulations demonstrate that this SDN architecture is effective and performs well in terms of costs, efficiency, and performance.

Keywords: SDN gateways, subnet abstraction, multi-class flow tables, rate adaptation, heterogeneous networks

OPEN ACCESS

Edited by:
Zhili Sun,
University of Surrey, United Kingdom

Reviewed by:
Hui Yang,
Beijing University of Posts and Telecommunications (BUPT), China
Xin Xu,
Army Engineering University of PLA, China

***Correspondence:**
Feng Liu
liuf@buaa.edu.cn

Specialty section:
This article was submitted to Aerial and Space Networks, a section of the journal Frontiers in Communications and Networks

Received: *31 May 2021*
Accepted: *27 August 2021*
Published: *14 September 2021*

Citation:
Sun J, Liu F, Li Y, Zhang L and Shi D (2021) A Software-Defined Architecture for Integrating Heterogeneous Space and Ground Networks. Front. Comms. Net 2:717476. doi: 10.3389/frcmn.2021.717476

1 INTRODUCTION

Nowadays, various networks have made considerable progress on their scales, types, technologies, and so on. Based on Cisco's annual internet report (Wallshein, 1999; Atm et al., 2017; Zhu et al., 2017; Hilvert-Bruce et al., 2018; Chen et al., 2019; Santos et al., 2019; McDowell 2020; Cisco Annual Internet Report, 2021), there will be 5.3 billion Internet users mobile users and 29.3 billion networked devices by 2023. Except for the rapidly growing scale of networks, a lot of new network applications emerge have emerged, e.g., short videos (Chen et al., 2019) and live streaming (Hilvert-Bruce et al., 2018). Besides, the most noticeable development of networks is reflected not only in ground networks, but also other types of networks, such as satellite networks (Zhu et al., 2017), the consultative committee for space data systems (CCSDS) testing and control (TTC) networks (Santos et al., 2019), datalink network (Wallshein, 1999), and aeronautical telecommunications networks

(ATNs) (Atm et al., 2017). For example, the Starlink system proposed by SpaceX has launched more than 1,300 satellites, and the final number of satellites will reach 12,000 (McDowell, 2020). The diversity of existing networks makes it possible to provide the integrated services across multiple networks.

The interconnection and integration of heterogeneous networks face many problems and challenges. There is significant heterogeneity between different types of networks. This heterogeneity is manifested in multiple aspects, i.e., the network architecture, network transmission protocol, dynamic and mobility support capability, coverage, data rate, switching mechanism, and quality of service (QoS) control technology (Miraz et al., 2017). First, different protocol architectures challenge the interconnection between heterogeneous networks. For example, The protocol architecture used in satellite networks mainly contains CCSDS and delay tolerant networks (DTN) (Roy et al., 2018). The protocol architecture researches of ground networks mainly focus on the transmission control protocol/Internet protocol (TCP/IP) architecture (Bohuslava and Martin, 2017). Second, the integration of heterogeneous networks should overcome their different switching mechanisms. For instance, the data link network is a time-division multiplexing network, and the forwarding of traffic flows depends on the allocated time slots (Wallshein, 1999). However, traditional TCP/IP networks are based on packet store-and-forward mechanisms. Third, different transfer modes also have effects on the integrating process of multiple networks. As an example, although both asynchronous transfer mode (ATM) and Internet protocol (IP) networks adopt the packet switching mechanism, ATM is connection-oriented and IP network is connectionless. Last, except for the aforementioned factors, there are other important factors that need to be considered in integrating heterogeneous networks, e.g., the dynamic and mobility support capability, coverage, data rate, resource capacity, and QoS control technology. For example, the rate adaptation should be performed among high-speed and low-speed networks, e.g., industrial Internet of things (IoT) and optical backbone networks (Wang et al., 2020). Therefore, all aforementioned factors make the integration of heterogeneous networks complex and non-trivial.

With the development of networks, existing network technologies are constantly evolving and new network technologies are constantly emerging. Since the appearance of the software-defined network (SDN) technology, it is considered a promising technology for interconnecting and integrating heterogeneous networks. SDN is a network architecture that includes the data plane, control plane, and application plane (Amin et al., 2018). The interacting interface between an SDN switch and an SDN controller is called the southbound interface. There are a number of existing southbound interface technologies, such as OpenFlow and P4Runtime (Kawaguchi et al., 2019). In our work, we consider OpenFlow as a typical southbound interface to design an SDN architecture that effectively integrates heterogeneous networks. Meanwhile, other southbound interface protocols can also be adopted in our proposed architecture using the similar methods. The interfaces between the control plane and the application plane are called northbound interfaces. The SDN application plane obtains the abstract model of underlying networks from the SDN control plane and passes the application requests in the SDN control plane through northbound interfaces, such as RESTful Application programming interface (API) (Arcuri, 2019). Besides, based on the northbound interfaces, network operators or even network users can customize their applications more conveniently. Therefore, a good line for solving the heterogeneity of various networks is to take full advantage of the hierarchical structures of SDN, which can decouple data plane and control plane, provide the abstract model of underlying networks, and enable the global optimization in networks.

Many research efforts have been devoted to interconnecting and integrating heterogeneous networks. First, traditional gateways are network elements serving as an access point to another network. There are gateways applied to various types of networks, such as the IoT (Zhu et al., 2010), wireless sensor network (Du et al., 2009), and home network (Saito et al., 2000). These traditional gateways are static protocol gateways and specific application-oriented. Their main functions are protocol conversion and data forwarding. However, these static protocol gateways need to be customized in advance for specific application requirements, which is a lack of dynamic flexibility. Second, traditional SDN gateways can connect multiple homogeneous SDN networks or traditional IP networks based on the access control and routing distribution strategies provided SDN controller (McKeown et al., 2008). However, traditional SDN gateways only support the interconnection of homogeneous networks, and only support the modification of header fields under the same protocol system, such as source network address translation (SNAT) and destination network address translation (DNAT) (Hoogendoorn (2021)). Last, many works improve the gateway mechanisms based on traditional SDN gateways to support the interconnection of heterogeneous networks. For example, some works improve traditional switches and routers to make them SDN-enable (Lin et al., 2016; Miano et al., 2017). However, the operations supported by traditional hardware switches are relatively simple, and complex operations cannot be completed, such as protocol conversion and rate adaptation. There are a number of works that utilize the P4Runtime technology to generate multi-class flow tables for parsing and deparsing different protocols of various networks (Uddin et al., 2018; Do et al., 2019). However, these works ignore that the integration of heterogeneous networks needs to take into consideration of not only protocol conversion but also other important factors, e.g., the switching mechanism, dynamic and mobility support capability, coverage, data rate, resource capacity, and QoS control technology. Some researchers use multiple SDN controllers to control different heterogeneous networks for realizing the interconnection of heterogeneous networks (Phemius et al., 2014; Blial et al., 2016). However, the interconnection methods using multiple SDN controllers can only achieve the integration of SDN-enable heterogeneous subnets. Li et al. 2019 proposed a simulation platform for software defined space-ground integrated network based on

Satellite Tool Kit (STK) and Mininet. Bertaux et al. 2015 proposed an SDN-enabled satellite and asymmetric digital subscriber line (ADSL) hybrid architecture. However, these works pay less attention to utilizing the global view of SDN controller for collaboratively optimizing multi-subnetworks. Therefore, for the heterogeneous network environment with various heterogeneous factors, it is necessary to design an efficient network architecture of interconnecting heterogeneous subnets.

In our work, we propose an SDN architecture interconnecting and integrating heterogeneous networks. The basic design principles of the integrated network architecture are to adopt SDN gateways to flexibly convert protocols and forward packets, utilize southbound interfaces to effectively shield the heterogeneity among subnetworks, and collaboratively optimize multiple subnetworks by abstracting the performance of each subnetwork. Our proposed architecture has the following contributions.

- First, we propose an efficient SDN gateway-based architecture of integrating heterogeneous space and ground networks (SD-SGN), and utilize southbound interfaces to effectively shield the heterogeneity among subnetworks. This SDN architecture based on can effectively reduce the SDN-enable costs of subsets and globally improve network performance.
- Second, in the SDN data plane, SDN gateways convert protocols and forward packets to interconnect heterogeneous networks. We propose multi-class multi-level flow tables to flexibly suit different protocol architectures. Besides, the rate adaptation of the proposed SDN data plane can reduce packet drops between subnetworks with different data rates and bandwidth capacities by specifying the forwarding rates and cache sizes of each flow.
- Third, in the SDN control plane, our proposed architecture abstracts the performance of each subnetwork to collaboratively generate flow tables based on optimization strategies, allocate resources based on subnet abstraction, and optimize the global network performance among multiple subnetworks. This effective approach can reduce the amount of statistical information needed to be collected, meanwhile, it can also save the bandwidth resources of transmitting collected data and the computing resources of processing collected data.
- Fourth, in the SDN application plane, our proposed architecture utilize SDN northbound interfaces to obtain the abstract models of each subnetwork and provide customizable and integrated network services, such as the services with low cross-network delays.
- Last, extensive simulation results show that our proposed SDN gateway-based architecture performs well in terms of the amount of collected data, SDN-enable costs, packet drops, and end-to-end delays across multi-subnetworks.

We organize the following paper as follows. **Section 2** introduces the state-of-the-art researches and existing problems. In **Section 3**, we propose the SDN gateway-based architecture of integrating heterogeneous networks and design each plane of the proposed SD-SGN architecture in detail. In **Section 4**, we perform extensive simulations and analyze the performance of proposed architecture from multiple aspects, i.e., the amount of collected data, SDN-enable costs, packet drops, and end-to-end delays across multi-subnetworks. **Section 5** concludes our work and points out the directions of future researches.

2 THE STATE-OF-THE-ART AND PROBLEMS

In this section, we give technical insights into the existing technologies that interconnect heterogeneous networks and point out the open issues that still need to be solved. Besides, we provide a detailed discussion on the technology trends of solving the integrating problem of heterogeneous space and ground networks.

2.1 The State-Of-The-Arts

Many research efforts have been devoted to integrating heterogeneous networks. Traditional gateways serve as access points to other networks. With the emergence of SDN technology, the flexible decoupling of the control plane and the data plane of SDN makes it possible to build integrated network applications. The existing interconnection technologies of different networks mainly cover the following categories:

Static protocol gateway: Traditional gateway is mainly for specific applications (Saito et al., 2000; Du et al., 2009; Zhu et al., 2010). The main functions of these application-oriented gateways are protocol conversion and data forwarding. They use static and fixed approaches of protocol conversion and routing strategies. However, static protocol gateways need to be customized in advance for specific application requirements, which is a lack of dynamic flexibility. When the state of subnets changes, the gateway parameters cannot be dynamically set and adjusted for dynamic adaptation between subnetworks.

Traditional SDN gateway: The SDN technology can decouple the functions of forwarding and control. Traditional SDN gateways perform access control and route distribution only according to the instructions and rules installed by SDN controllers to connect homogeneous SDN networks with IP networks, or homogeneous SDN networks with IP networks (McKeown et al., 2008). However, traditional SDN gateways only support the interconnection of homogeneous networks. Besides, they only support the modification of header fields under the same protocol architecture. Traditional SDN gateways often neglect the inconsistency between heterogeneous networks, such as communication modes, transmission mechanisms, protocol architectures, resource capabilities, and other characteristics. Therefore, traditional SDN gateways are not suitable for interconnection and cooperative control between heterogeneous subnets.

Improved SDN gateways: Many works focus on improving the gateway mechanisms based on traditional SDN gateways to support the interconnection of heterogeneous networks. These improved SDN gateways can be classified further into the following kinds. First, some works utilize SDN-supported middlewares to make traditional forwarding devices (e.g., routers and switches) SDN-enable. It can convert traditional forwarding devices into SDN switching nodes by installing SDN middlewares (Miano et al., 2017). Then, SDN controllers can connect to and control these SDN-enable devices through southbound interface protocols so that general IP networks can be controlled in SDN mode. However, traditional hardware forwarding devices based on network and data link layers are designed for access control and routing distribution. These devices can support simple operations, which are difficult to accomplish complex protocol conversions, rate adaptations, etc. Therefore, traditional hardware forwarding devices with SDN-supported middlewares can not well realize the interconnection between heterogeneous networks. Second, flow tables supporting multi-class protocols and single pipeline structures can realize protocol conversion of multiple heterogeneous subnets. In traditional SDN switches, the original single pipeline can be inserted with flow tables of multiple protocols to support the matching and interconnection of multiple heterogeneous subnets. However, the structure that multi-class protocol flow tables are arranged into a pipeline results in more levels of flow tables, which increases the time of look-up tables. Third, the SDN gateways supporting border gateway protocol (BGP) can realize the interconnection between SDN networks and traditional non-SDN IP networks (Lin et al., 2016). Some works are to install BGP protocol modules in SDN gateways and regard the whole SDN network as a router of traditional IP networks. Traditional IP networks and SDN networks interact with network layer reachability information (NLRI), so as to realize the interconnection between SDN networks and non-SDN traditional IP networks. However, the SDN gateways supporting BGP protocols can only connect SDN networks and traditional IP networks and do not support the interconnection with other networks with non-IP switching mechanisms. Fourth, the SDN gateways supporting static protocol conversion can parse and deparse multi-class protocols of different subnetworks. The static protocol conversion modules can be installed in SDN gateways to convert all types of protocols into the same common protocol format, which makes multi-class protocols share the same type of flow table. After leaving the SDN gateway, the traffic flow is transformed into the protocol format of the target subnet through static protocol conversion, so as to realize the interconnection between heterogeneous networks. However, SDN gateways supporting static protocol conversion need to convert all types of network protocols into the same type of protocol, which increases the complexity of protocol conversion. In addition, this method needs to be customized in advance for specific application requirements and lacks dynamic flexibility. Fifth, some works utilize the P4Runtime technology to generate multi-class flow tables for parsing and deparsing different protocols of various networks (Uddin et al., 2018; Do et al., 2019). However, these works ignore other heterogeneous factors of different subnetworks. For example, the integration of multiple subnetworks with different switching mechanisms and resource capacities needs not just protocol conversion but also rate adaptation and cache allocation. Sixth, the integration of heterogeneous subnets can be realized by the cooperative control of multiple SDN controllers (Phemius et al., 2014; Blial et al., 2016). To shield the heterogeneity of the subnets, each heterogeneous subnet should be controlled by an SDN controller individually, and these controllers share the subnetwork information collected from each subnet through eastbound and westbound interfaces. When a traffic flow is transmitted across heterogeneous networks, multiple SDN controllers plan the service transmission path cooperatively, and each controller installs OpenFlow rules into SDN switches in their controlled subnetworks, respectively. However, this method needs to make each subnet SDN-enable in advance, which brings high SDN-enable costs. Besides, this approach does not support the interconnection with the non-IP switching networks or the networks unable to be SDN-enable. The typical scenarios of heterogeneous network interconnection and integration are depicted in **Figure 1**, and the heterogeneous subnetworks of the integrated network are shown in **Figure 2**.

2.2 Technical Trends

The network heterogeneity poses a severe challenge to the interconnection and integration of multiple types of networks. However, the future integrated networks interconnecting the inconsistent underlying network should provide a unified centralized control plane to ensure the resource allocation and collaborative optimization of the integrated networks. With the emergence of SDN technology, the idea of forwarding and control separation provides a new way to realize the interconnection and integration of heterogeneous networks.

Southbound interfaces and shielding heterogeneity: The basis of collaborative control of various networks that have different technology systems is a common abstract network model. The implementation of network abstraction needs to be supported by SDN southbound interfaces. Based on the subnet state information collected by southbound interfaces, the SDN controller can establish a unified network model for each subnet to shield networks' heterogeneity. When the integrated network application manages or uses networks, logic modules in SDN controllers can automatically achieve the protocol conversion, rate adaptation, and resource coordination through southbound interfaces, without caring about the details of underlying networks. The southbound interface technology makes it possible to use the global view of the SDN controller to automatically adapt the protocols and bandwidth between heterogeneous networks, coordinate the resources between heterogeneous networks, and flexibly build integrated network applications.

Multi-class multi-level flow tables and protocol conversion: In the data plane of SDN, the architecture of integrated networks should use SDN gateways to connect heterogeneous subnets to avoid the high costs of making each subnet SDN-enable. The existing SDN flow tables should be improved so that the SDN gateways can flexibly maintain the flow tables of different protocols of various subnets. Besides, the improved flow tables should be able to allocate the cache resources and specify the forwarding rates according to the subnet performance, so as to realize the rate adaptation between subnets with different performance.

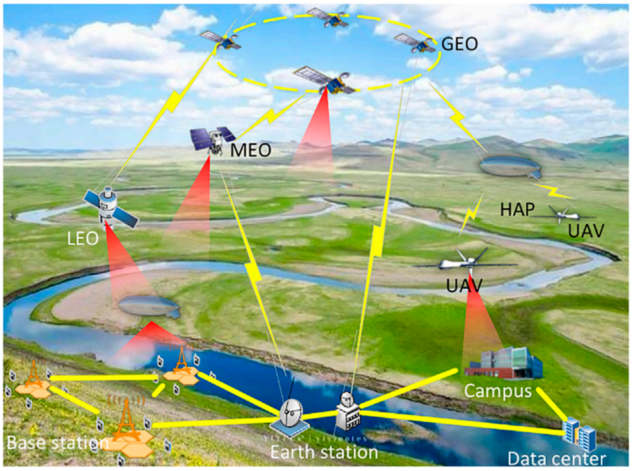


FIGURE 1 | Typical scenarios of heterogeneous network interconnection and integration based on space, air, and ground networks.

Subnetwork abstraction and global optimization: Based on the estimated subnet performance, the gateway-level paths are allocated for each traffic flow to realize the subnet selection for cross-network transmission and the cooperation between heterogeneous subnets. The complicated intra-subnet path planning is still controlled by the traditional switching and routing devices inside each subnet. Therefore, the cross-subnet routing can be loop-free, when the proposed control plane adopts traditional routing strategies, e.g., Dijkstra. This architecture can effectively avoid the long convergence time of routing calculation, the long delays of instruction packets transmission, and the calculation bottleneck of SDN controllers.

SDN-based network architecture and heterogeneous network integration: For the heterogeneous network scenarios with heterogeneous protocols, dynamic topology, and diverse switching mechanisms, it is necessary to design a new network architecture of heterogeneous subnetwork interconnection. This architecture should be able to establish a unified abstract network model for each subnet based on SDN southbound interfaces to shield the heterogeneity of subnetworks. Through the SDN southbound interfaces, the control information calculated by SDN controllers and the network state information collected by SDN gateways interact in the secure socket layer and transport layer security (SSL/TLS) links (Radivilova et al., 2018). The integrated network architecture can realize the interconnection and integration of heterogeneous networks, as well as the integration of transmission, processing, and application services.

3 SOFTWARE-DEFINED-SPACE AND GROUND NETWORKS: ARCHITECTURE OVERVIEW AND SYSTEM DESIGN

The existing interconnection technologies are not suitable for heterogeneous space and ground networks with dynamic protocol conversion, node mobility, and limited resources. As for the application-oriented static protocol conversion gateways, their underlying subnets are inconsistent and their upper layer is distributed protocol conversion. For traditional SDN gateways, the processed underlying subnets are consistent with each other, and their upper layer is unified and centrally controlled. Future integrated networks need various types of underlying networks and provide a unified and centralized control layer to ensure the resource allocation and collaborative optimization of the integrated networks.

3.1 Architecture Overview

To tackle the aforementioned issues, we propose an SDN network architecture for interconnecting heterogeneous networks, which utilizes southbound interfaces to obtain a unified abstract model. As shown in **Figure 3**, the proposed SD-SGN architecture consists of the data plane, and control plane, and application

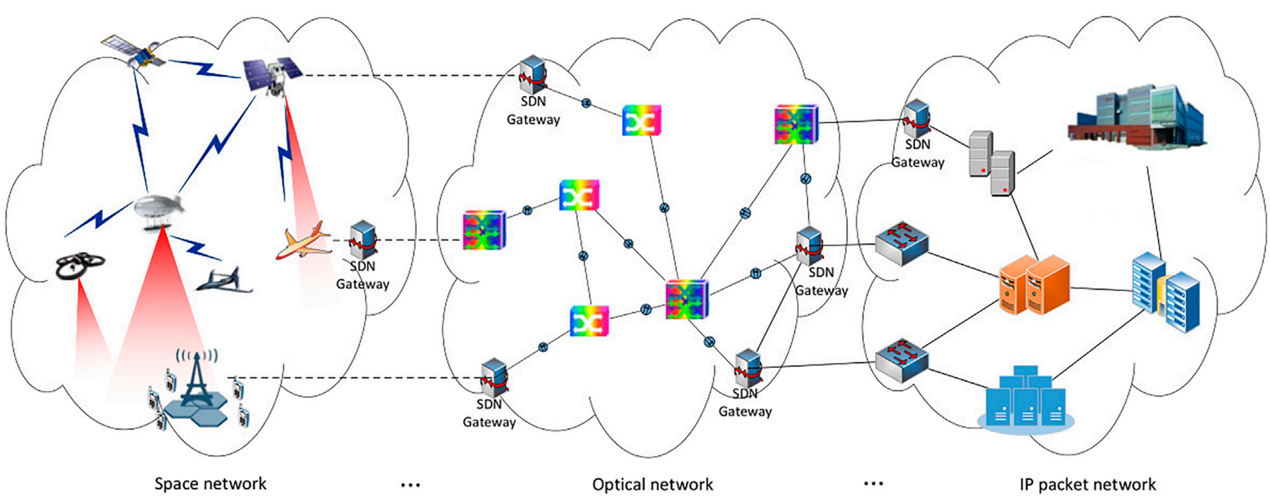


FIGURE 2 | The heterogeneous subnetworks of the integrated network including space networks, optical networks, IP packet networks, etc.

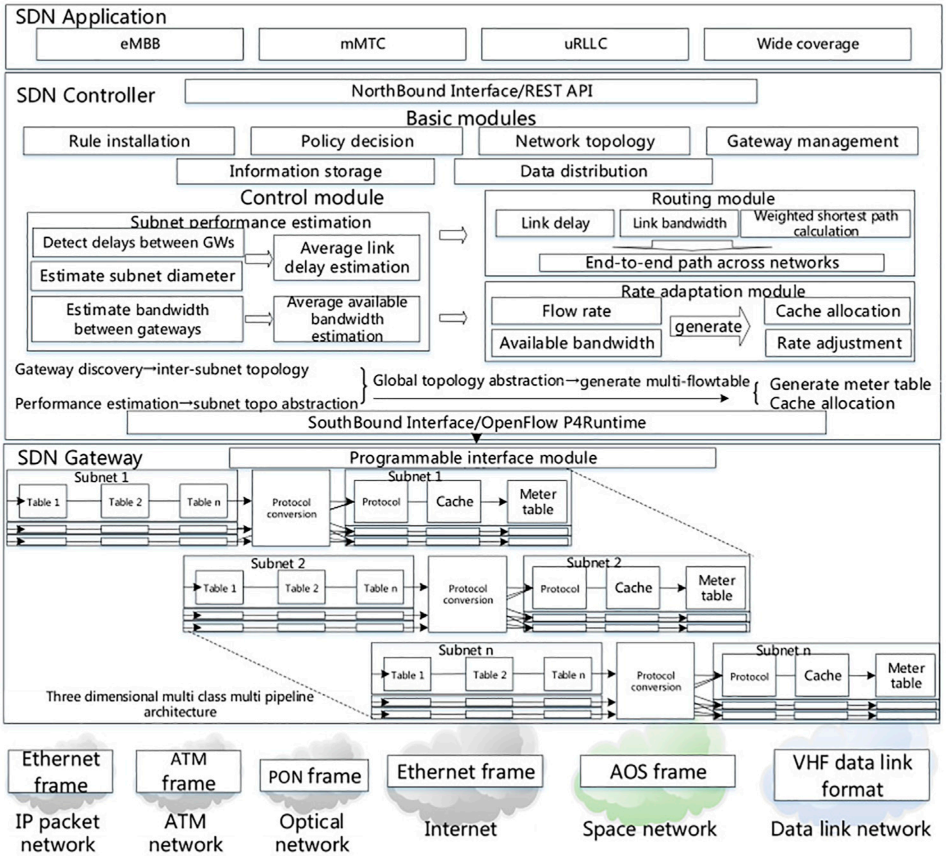


FIGURE 3 | The implementation of the SD-SGN architecture realizing the interconnection of heterogeneous networks.

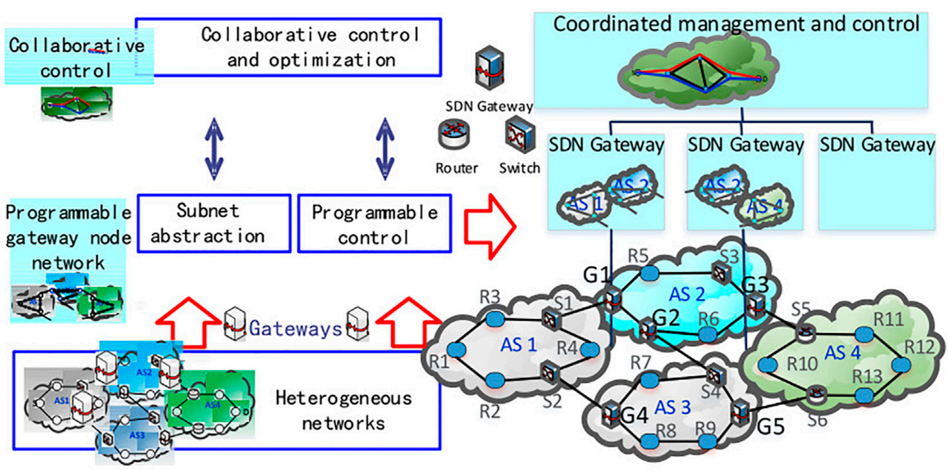


FIGURE 4 | The SD-SGN architecture interconnects and integrates of heterogeneous networks.

plane. Note that the data plane of SD-SGN architecture consists of SDN gateways with multi-class multi-level flow tables instead of traditional SDN switches. SDN southbound interfaces shield the complex packet operations of connecting heterogeneous subnet on the data plane, which includes protocol conversion, packet distribution, rate adaptation, etc. In the control plane, SDN southbound interfaces are only used for collecting and processing the SDN gateway information to abstract the topology and performance information within and between subnetworks. This architecture realizes the decoupling of control and forwarding functions in heterogeneous interconnected networks.

In the data plane of SD-SGN architecture, this architecture designs a flow table with a multi-class and multi-level structure to realize flexible and efficient processing of data flows in a heterogeneous network. Compared with flow tables with a single-type single-pipeline structure in the traditional SDN gateways, the multi-type multi-level structure changes the flow table structure from two-dimensional to three-dimensional. The three-dimensional flow table is not only suitable for the matching and operation of heterogeneous network protocols, but also reduces the number of flow tables in a single pipeline and reduces the delays of flow table matching. In addition, according to multi-type and multi-pipeline flow tables, different buffer sizes and forwarding rates are assigned to the forwarding queues of different subnets, so as to realize the integration and interconnection of heterogeneous subnets with different performance.

In the control plane of SD-SGN architecture, this architecture designs a global network abstraction based on southbound interface and SDN gateways and proposes an effective SDN controller to generate the control strategies of SDN gateways. Compared with the traditional SDN controllers that collect the complete topology and generate flow tables of SDN switches, the SD-SGN architecture not only saves the SDN-enable costs of each heterogeneous subnetwork but also globally optimize the performance of multiple networks. The SD-SGN architecture is from a perspective of subnetwork abstraction to generate routing paths between SDN gateways. These methods make the control plane focus on controlling the cooperation between subnets. The data transmission within the subnets is still controlled by the original switching and routing mechanism in these subnets. In addition, this control plane also avoids the controller's calculation bottleneck and the SSL/TLS links' transmission bottleneck.

The improved data plane and control plane of the proposed SD-SGN architecture interact through the SDN southbound interface. The SD-SGN architecture realizes the interconnection of heterogeneous networks to form an integrated network.

3.2 Integrated Network Architecture Design

The SD-SGN architecture establishes a unified SDN network model abstraction for each subnet based on the southbound interfaces to shield the heterogeneity of various networks. SDN southbound interfaces separate the packet operations from the control plane. These complicated operations are handled by the multi-class multi-level flow tables in the data plane. As shown in **Figure 4**, SDN southbound interfaces only collect data from SDN gateways to abstract the global topology and subnet performance. The left part of **Figure 4** shows that the control plane of the SD-SGN architecture can abstract the subnet performance and provides the programmable control of SDN gateways. In the environment of heterogeneous subnet interconnection, the SD-SGN architecture implementing SDN gateways with multiple types of flow tables and SDN controllers that abstract heterogeneous subnet information can effectively solve the issues in heterogeneous subnets, e.g., address mapping, routing calculation, and rate adaptation. Compared with static protocol gateways and traditional SDN gateways, the SD-SGN architecture decouples the control and forwarding functions in heterogeneous networks and realizes the flexible and effective interconnection of heterogeneous networks.

3.3 Data Plane Design

In the data plane, the SD-SGN architecture design the flow table structure with multiple types and multiple pipelines to achieve flexible and efficient processing of traffic flows across heterogeneous networks.

SDN gateway-based subnetwork interconnection: The data plane of SD-SGN architecture utilizes SDN gateways to connect heterogeneous networks. The SDN gateway-based interconnection not only ensures the flexibility of packet processing but also be easily realized without making each subnet SDN-enable. As shown in **Figure 4**, there are four heterogeneous networks of AS1, AS2, AS3, and AS4. They are connected by five SDN gateways. The forwarding devices in these subnetworks are still traditional switches and routers. Compared with the interconnection scheme of multiple SDN controllers, the SD-SGN architecture can save the SDN-enable costs of each subnetwork. Meanwhile, for those heterogeneous subnets that are non-IP mechanisms or are hard to be SDN-enable, the SDN gateway-based architecture is a flexible and appropriate heterogeneous interconnection solution.

Flow tables with multi-class multi-level structure: The SD-SGN architecture designs a flow table structure of multiple types and multiple pipelines. By increasing the dimension of subnet types, the SDN gateway generates a pipeline of flow tables for each subnet, and the traditional flow table structure is changed from two-dimensional to three-dimensional. For various types of subnet protocols, different protocol fields have different meanings. Therefore, the processing of each type of subnet protocol needs a separate parsing and deparsing method, i.e., using multi-class flow tables to simplify the parsing and deparsing of packets. The SDN gateways in SD-SGN architecture need to maintain flow table entries and protocol conversions in the flow table structure of multiple types and multiple pipelines. Besides, the flow tables should be divided into multiple tables and organized in a pipeline mode to save memory resources in SDN gateways. As shown in **Figure 4**, the SDN controller determines the types of subnets connected to each switch port of gateways G1, G2, G3, G4, and G5, according to the collected protocol information. For example, the SDN controller installs the protocols of AS1 and AS2 into the gateway G1. If a traffic flow's source and destination nodes are R1 and R12 and it goes through AS1, AS2, and AS4, respectively, the matching table of this flow is installed in G1 and G3. After that flow matches a table entry in G1, its packet header is converted from the AS1 type to the AS2 type, and its payload is encapsulated and passed in the AS2 subnet. The maintenance of different types of flow tables on each SDN gateway does not affect each other. Besides, compared with the two-dimensional flow tables of the traditional SDN gateway, the multi-class multi-level structure can effectively reduce the packet matching delays in flow tables.

Forwarding rate adaptation and cache size allocation Except for the protocol conversion based on the SDN flow tables, the SD-SGN architecture also utilizes the SDN meter table to control and adjust the forwarding rates of different traffic flows Šeremet and Čaušević (2020). Besides, the proposed gateway should flexibly modify the cache queue sizes of each meter table to dynamically cache the accumulated data due to the performance differences of subnets. When the traffic rate of a flow is larger than the available bandwidth of a subnet hosting that flow, the SDN gateway allocates the appropriate cache sizes and specify the appropriate forwarding rates. The SDN controller estimates the average available bandwidth of each link of the subnets that use packet transmission mechanisms or the time average available bandwidth of the subnets that use time-division transmission mechanisms. In **Figure 4**, the estimated available bandwidths of AS2 is $b2$. A traffic flow f traversing from AS1 to AS2, its traffic rate b_f is larger than $b2$, and its duration is t seconds. In the SDN gateway G1, its specified forwarding rate b_f^1 should be smaller than $b2$, and its allocated cache size should be equal to the product of duration and the difference of rates, i.e., $t(b_f - b_f^1)$. Compared with static protocol gateways and the SDN gateways improved by static protocol conversion, the SD-SGN architecture can not only avoid the repeated and complicated operations of protocol conversion but also dynamically set and adjust the gateway parameters to achieve performance adaptation between subnetworks.

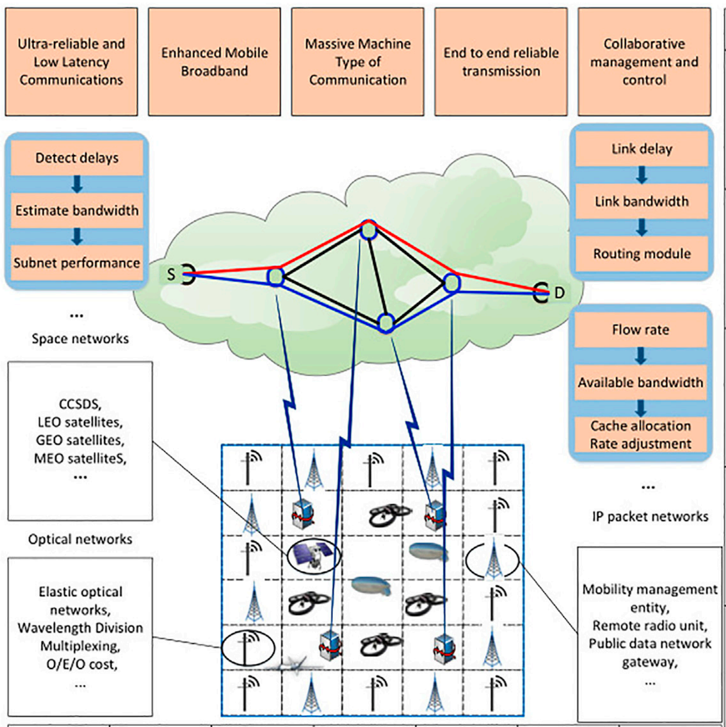
3.4 Control Plane Design

In the control plane, the SD-SGN architecture can provide a global network abstraction based on SDN southbound interfaces and SDN gateway information, and design SDN controllers generating the control strategies of SDN gateways.

Subnetwork performance estimation and cross-network routing: In the SD-SGN architecture, the SDN controllers estimate the performance of subnetworks based on SDN gateways to routing calculation and rate adaptation across networks. By allocating cache resources and specifying forwarding rates, the SD-SGN architecture can balance the performance differences of heterogeneous networks hosting the same traffic flows. The estimated network performance mainly includes network diameters, average link delay, and average available bandwidth. As shown in **Figure 4**, there are several SDN gateways in the subnetwork AS2. To estimate the performance of the subnetwork AS2, the SDN controller sends the AS2 performance detection instructions to G1, G2, and G3, respectively. Take G1 for example, the gateway G1 floods its performance detection packets in subnet AS2. When the performance detection packets visit G2 and G3, because these packets can not be recognized, they are passed in the SDN controller. Then, the SDN controller can analyze the changes of time to live (TTL) fields of these packets to obtain the hop counts between G1 and G2 and the hop counts between G1 and G3 (Giovane et al., 2019). This operation is repeated for G2 and G3 to obtain the hop counts among these SDN gateways. The average distance between SDN gateways is considered as the estimated network diameter of AS2. Using the *ping* command to obtain the round-trip time (RTT) between each pair of SDN gateways. We use the result of dividing the average RTT by the estimated network diameter to estimate the average link delay of AS2. Based on the performance detection information obtained from SDN gateways, the SDN controller can obtain the average link available bandwidth using bandwidth estimation algorithms, such as TCP Westwood (Haveliwala et al., 2020) and TCP Vegas (Chowdhury and Alam, 2019). This SDN gateway-based architecture can not only avoid the high cost of making subnets SDN-enable but also avoid the long convergence time collecting the complete network information.

Subnetwork abstraction and global network model: The SD-SGN architecture utilizes southbound interfaces and SDN gateways to abstract the subnet performance and the global topology. Through collecting gateway information, this architecture can obtain the topology of inter-subnets. The SDN controller can estimate the subnetwork performance by collecting the detection information among SDN gateways. Based on the abstract global topology, the SDN controller calculates the routing paths consisted of SDN gateways for traffic flows, so as to collaboratively control multiple heterogeneous subnets. The abstraction capability of the global network enables SDN controllers to control the more important cooperation between subnets from a higher perspective of subnet abstraction. As depicted in **Figure 4**, the SDN controller abstracts each subnet into a subnet node in the network, i.e., the nodes of AS1, AS2, AS3, and AS4. Besides, the SDN gateways are considered as the links between subnet nodes. For example, the gateways G1, G2, and G3 are considered as the links for connecting the AS2 node with the AS1, AS2, and AS3 nodes, respectively. The estimated performance is considered as the parameters of each subnet node, which include the network diameter, available bandwidth, and link delay. This abstraction model enables the SDN controller to control the collaboration between subnets with low costs and achieves a scalable interconnection and convergence of heterogeneous networks. Compared with traditional SDN gateways, the SD-SGN architecture can reduce the SDN-enable costs, the amount of collected data, and the convergence time of cross-subnet routing.

Decoupling of logical control and data processing: Based on the idea of function separation, the SD-SGN architecture decouples logical control and data processing, i.e., the functions of address calculation, routing calculation, and rate adaptation are placed on the SDN controller, and the functions of protocol conversion, data forwarding, and flow rate adjustment are placed on the SDN gateway. This architecture eliminates the dependence on the SDN controllers for packet operations of traffic flows across networks. Compared with the SDN gateways using SDN controllers to convert protocols, the SD-SGN architecture avoids the computing bottleneck of the SDN controllers and the bandwidth bottleneck of links connecting to the SDN controllers. Therefore, the proposed architecture can achieve the scalable interconnection and integration of heterogeneous networks.



	Number	Traffic rate	Across networks	Source subnet	Destination subnet	Time of duration
Traffic flows	Unif(10,100)	Unif(50,950)	Unif(1,20)	Unif(1,20)	Unif(1,20)	Unif(100,1000)

A

Parameter settings	Networks	Values
Average SDN-enable costs	IP packet network	Unif(10,20)
	ATN network	Unif(30,40)
	Optical network	Unif(50,60)
	Space network	Unif(400,600)
	Data link network	Unif(30,50)
	ATM network	Unif(20,40)
The number of nodes	IP packet network	Unif(100,200)
	ATN network	Unif(300,400)
	Optical network	Unif(500,600)
	Space network	Unif(40,60)
	Data link network	Unif(300,500)
	ATM network	Unif(200,400)
Average memory resources	IP packet network	Unif(100,200)
	ATN network	Unif(300,400)
	Optical network	Unif(450,500)
	Space network	Unif(40,60)
	Data link network	Unif(300,500)
	ATM network	Unif(200,400)
Average bandwidth resources	IP packet network	Unif(1000,2000)
	ATN network	Unif(3000,4000)
	Optical network	Unif(5000,6000)
	Space network	Unif(400,600)
	Data link network	Unif(3000,5000)
	ATM network	Unif(2000,4000)
Average link delays (ms)	IP packet network	Unif(10,20)
	ATN network	Unif(30,40)
	Optical network	Unif(50,60)
	Space network	Unif(400,600)
	Data link network	Unif(30,50)
	ATM network	Unif(20,40)

B

Average information collection (Kbit)	SDN gateway	Unif(30,60)
	SDN switches	Unif(5,15)

C

FIGURE 5 | Application case study and performance quantitative analysis of the SD-SGN architecture.

3.5 Application Plane Design

Based on the SDN northbound interfaces, the application plane of the SD-SGN architecture can obtain the global abstract network model of heterogeneous networks from the SDN control plane, and send the network requests to the SDN control plane. The network operators or users can conveniently customize their network services without caring about the implementation details of underlying networks. The SD-SGN architecture integrating various heterogeneous networks can provide integrated transmission and processing applications with high bandwidth, low latency, large capacity, and wide coverage. These integrated network applications have important values in the fields of civil applications, national security, disaster warning, etc. As shown in **Figure 4**, the application plane customizes a low-latency and high-reliability network service for a traffic flow f that has low duration and volume. The source node and destination node of flow f are in the subnets AS1 and AS4, respectively. As the middle subnets, the estimated diameter and the estimated average link delay of AS2 are smaller than the ones of AS3, and the estimated available bandwidth of AS3 is larger than the one of AS1. Based on the global abstract network model obtained from the SDN plane, the application plane specifies that the flow f needs to go through AS1, AS2, and AS3. Then, the control plane installs flow tables for f in G1 and G3, and specifies the corresponding forwarding rate and cache size. This architecture can utilize the SDN northbound interfaces to effectively provide network applications across heterogeneous subnets.

4 CASE STUDY AND QUANTITATIVE ANALYSIS

In this section, we provide the cases the SD-SGN architecture applied in and given a comprehensive quantitative analysis of the architecture performance, which includes the SDN-enable costs, the amount of collected data, the amount of dropped packets, and the total end-to-end delay. Among them, to make a network SDN-enable, the network operator needs to use SDN switches and SDN gateways to replace the traditional network devices. The SDN-enable costs mainly refer to the Operational Expenses (OPEX) and Capital Expenditures (CAPEX) spent in purchasing, managing, and maintaining these SDN devices. In the simulation section, we omit the unit of SDN-enable costs to cover different types of SDN-enable costs, e.g., the employee's working hours.

4.1 Network Scenarios

To evaluate the performance of the SD-SGN architecture, we carefully set the parameter settings of heterogeneous networks in

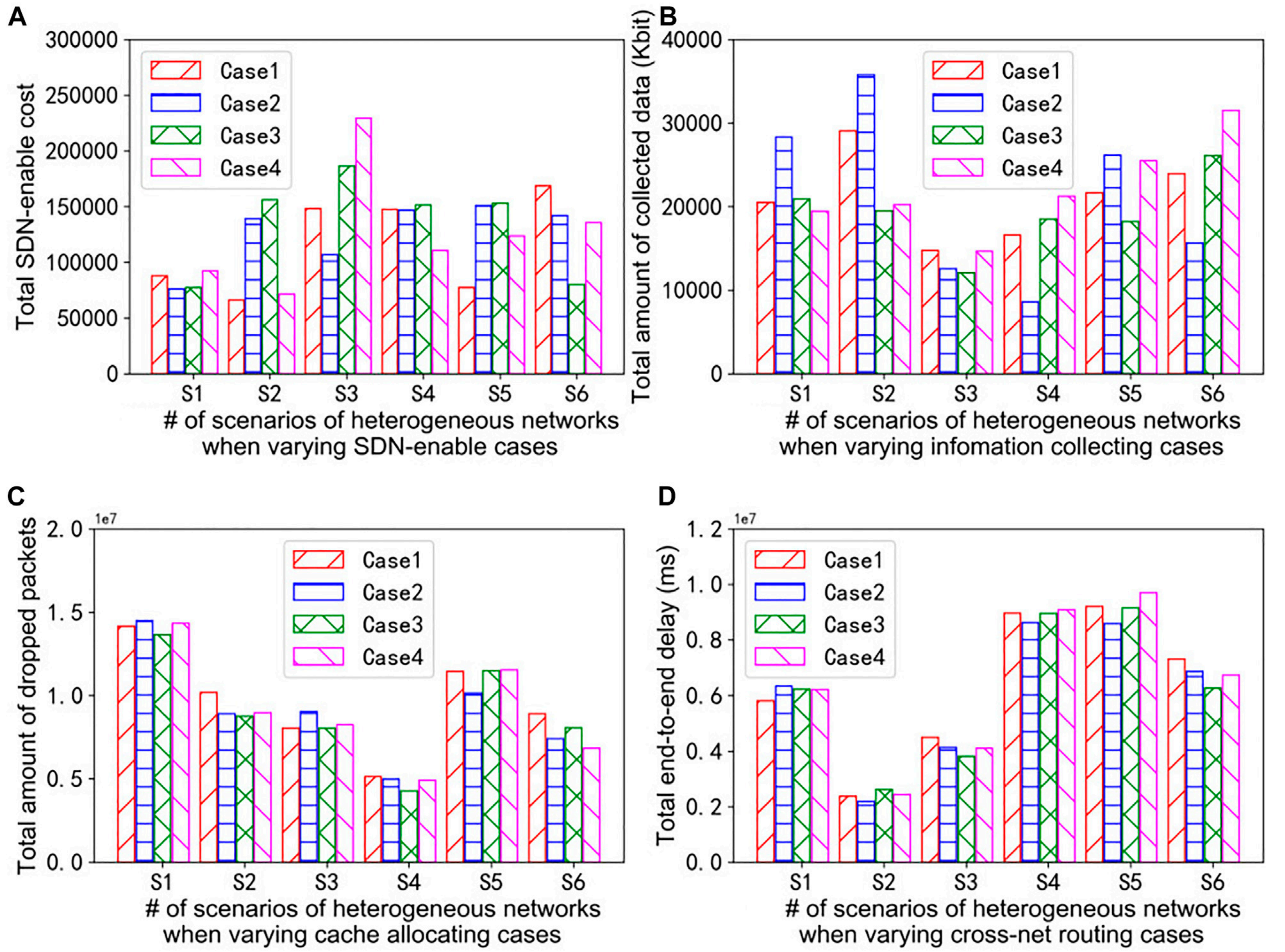


FIGURE 6 | Performance evaluation: **(A)** the SDN-enable costs; **(B)** the amount of collected data; **(C)** the amount of dropped packets; **(D)** the total end-to-end delay.

the simulation experiments. As shown in **Figure 5A**, in our simulations, we mainly focus on the parameters that have effects on the performance of the integrated network. First, we choose six types of networks with obvious heterogeneity, which include IP packet networks, ATN networks, optical networks space networks, data link networks, and ATM networks. Some heterogeneous characteristics of these networks are summarized in **Figure 5A**. There are various network applications that can be constructed based on these types of networks, such as ultra-reliable and low latency communications (uRLLC), enhanced mobile broadband (eMBB), and machine-type communications (mMTC) (Popovski et al., 2018). Second, the topologies of each subnetwork are randomly generated connected graphs, the connection relationship among these subnetworks is random, and the gateways are randomly connected to the edge switches. Third, according to the uniform distribution, the average SDN-enable costs of these six networks are randomly set within the domains of [10,20], [30,40], [50,60], [400,600], [30,50], and [20,40], respectively. Fourth, the number of nodes of IP packet networks follows the uniform distribution and ranges from 100 to 200. The ones of ATN networks, optical networks space networks, data link networks, and ATM networks are from 300 to 400, from 500 to 600, from 40 to 60, from 300 to 500, and from 200 to 400, respectively. Fifth, we randomly set the average memory resources of gateways of IP packet networks within the range from 100 to 200 according to the uniform distribution. The ones of five other types of networks are set within the ranges of [300,400], [450,500], [40,60], [300,500], and [200,400], respectively. Sixth, we set the value ranges of the average bandwidth resources of these six networks as [1000,2000], [3000,4000], [5000,6000], [400,600], [3000,5000], and [2000,4000], respectively. Seventh, the average link delays of IP packet networks, ATN networks, optical networks space networks, data link networks, and ATM networks are randomly set from the ranges of [10,20], [30,40], [50,60], [400,600], [30,50], and [20,40], respectively. Eighth, the amount of collected information from SDN gateways and SDN switches are within [30,60], and [5,15], respectively. Last, as for traffic flows, their source nodes and destination nodes are in random subnetworks, their traffic rates range from 50 to 950, and their time of duration is randomly set within [100,1000]. The detailed parameter settings are summarized in **Figure 5**.

4.2 Performance Evaluation

In this subsection, we carefully design the simulation experiments based on the proposed SD-SGN architecture and evaluate the performance improvement from multiple aspects, i.e., the SDN-enable costs, the amount of collected data, the amount of dropped packets, and the total delays. The simulation experiments discussed below are carried out on MATLAB and conducted in the server with a 2.10 GHz Intel Xeon processor and 64GB RAM.

The SDN-enable cases and total SDN-enable costs: We randomly generate six network scenarios according to the parameter settings listed in **Figure 5**. These network scenarios are denoted as S1, S2, S3, S4, S5, and S6, respectively. A subnetwork can be SDN-enable using traditional SDN architecture or the SD-SGN architecture. Traditional SDN architecture needs to make each node in the network SDN-enable. The latter is an SDN gateway-based architecture where the subnets are still controlled by the original switching and routing mechanisms. We randomly generate four SDN-enable cases of Case1, Case2, Case3, and Case4 for each network scenario. As shown in **Figure 6A**, the costs of four cases for making S3 SDN-enable are 148162, 106947, 186393, and 229624, respectively. Compared with Case1, Case3, and Case4, Case2 has the lowest SDN-enable cost. This is because most of the subnetworks in Case3 adopt the SD-SGN architecture, i.e., these subnetworks only need to make the gateway SDN-enable. The average SDN-enable cost in S1, S2, S3, and S4 are 83411.5, 108243.25, 167781.5, 139100.25, 126282.5, and 131580.5, respectively. The average SDN-enable cost of S3 increases 101.2% compared with the value of S1. This result may be resulted by the larger scale of S3, i.e., the number of subnets in S3 and the number of nodes of each subnet in S3 may be larger than the ones in S1. Besides, the average SDN-enable costs of four cases are 115994.6, 126997.2, 134090, and 127184.5, respectively. The cases with lower costs may prefer to use the SDN gateway-based methods to make the subnets SDN-enable, whose SDN-enable cost is significantly less than that of the traditional SDN architecture.

Information collecting cases and total amount of collected data: We randomly generate six network scenarios and four collecting cases used for evaluating the amount of collected data of different SDN architectures. For traditional SDN architecture, the controller needs to collect state information of all SDN switches in subnets to construct a complete topology. However, the SD-SGN gateways can utilize the information from SDN gateways to provide a global abstract network model, which needs a less amount collected data. As shown in **Figure 6B**, the amount of collected data of four cases in scenario S2 are 29035, 35774, 19507, and 20221, respectively. Case2 and Case3 obtain the largest and smallest amount of collected data, respectively. This may be caused that more subnets in Case1 adopt the traditional SDN architecture that needs to collect state information of all SDN switches. Case3 constructs the global network model by utilizing the SD-SGN architecture to collect the network information, which only needs to collect the data from SDN gateways. Besides, because of various network scales of generated scenarios, the average amount of collected data of these network scenarios are different, i.e., the ones of six scenarios are 22289.2, 26134.2, 13531.5, 16242, 22876, and 24298.7, respectively.

Cache allocating cases and total amount of dropped packets: To explore the effects of cache allocation on network performance, we generate network scenarios and traffic flows based on the parameters listed in **Figure 6**. We also randomly specify four cache allocating cases. For the same network, various cache allocating schemes have a different amount of dropped packets, e.g., the total amount of dropped packets of four cases in S6 are 8898766, 7412836, 8075736, and 6827660, respectively. Besides, for various scales of network scenarios, the total amount of dropped packets are also different. For Case1, its dropped packets in these scenarios are 14154859, 10186956, 8043071, 5131715, 11442115, 8898766, respectively. The total amount of dropped packets of all cases in these six network scenarios are 56629383, 36808328, 33371859, 19289540, 44600497, 31214998, respectively.

Cross-network routing cases and total end-to-end delays: Based on the parameter settings shown in **Figure 6**, we randomly generate the network scenarios and traffic flows to explore the changes of end-to-end delays when varying cross-subnet routing cases. It is hard to coordinate multiple networks to optimize cross-network traffic flows based on traditional gateways. Traditional SDN gateways often ignore the effects of subnetwork performance on cross-network traffic flows, which includes network diameters, average link delay, and average available bandwidth. As shown in **Figure 6D**, the total end-to-end delays of four routing cases for S6 are 7312217, 6866587, 6273639, and 6733242, respectively. The obtained end-to-end delay of Case3 is lowest because Case3 adopts the SD-SGN architecture to route traffic across networks. Based on the data collected from SDN gateways, the SD-SGN architecture can estimate the performance of each network and abstract the global network model. When routing cross-subnet traffic flows, the SD-SGN architecture not just consider the number of subnets needed to go through, but also the scale and performance of each subnet hosting the flows. Therefore, the SD-SGN architecture can reduce the end-to-end delays of flows across networks. Besides, for Case3, the obtained end-to-end delays of six cases are 6229391, 2619354, 3810341, 8955832, 9159855, and 6273639, respectively. This is because the end-to-end delays are affected by the varying parameters of networks and flows, such as the subnets of source nodes, the subnets of destination nodes, and the connection relationship between subnets.

The obtained results depicted in **Figure 6** indicates that the SD-SGN architecture outperform other solutions for interconnecting and integrating heterogeneous networks. **Figure 6A** shows that the unified architecture based on SDN gateways can reduce the SDN-enable costs of subsets. As shown in **Figure 6B**, the rate adaptation of the proposed SDN data plane can reduce packet drops between subnetworks with different data rates and bandwidth capacities by specifying the forwarding rates and cache sizes of each flow. In **Figure 6C**, the SDN control plane of our proposed architecture abstracts the performance of each subnetwork based on SDN gateways, which can reduce the amount of statistical information needed to be collected. **Figure 6D** indicates that the application plane of the SD-SGN architecture can utilize the abstract models of each subnetwork to provide the services with low cross-network delays. Except for above evaluations, we can also provide a qualitative analysis on other criteria of the proposed architecture, e.g., throughput. The control plane of our proposed architecture can estimate the performance of each network and abstract the global network model. When a congested subnetwork appears in the shortest path between the source and destination nodes, the control plane of our proposed architecture can detect the average available bandwidth of this subnetwork based on the collected data, and guide the traffic flows to go through other subnetworks with more bandwidth resources. Therefore, the network throughput can be improved effectively using our architecture.

5 CONCLUSION

In our work, we propose an efficient SDN gateway-based architecture of integrating heterogeneous space and ground networks. In the control plane, the SD-SGN architecture can abstract the performance of each subnetwork to collaboratively optimize the global network performance among multiple subnetworks. In the data plane, the SD-SGN architecture utilizes multi-class multi-level flow tables to flexibly suit different protocol architectures and specifies the forwarding rates and cache sizes to perform the rate adaptation between subnets with different performance. In the application plane, the SD-SGN architecture can utilize the global abstract network model to provide customizable and integrated network services. Extensive simulation results show that the SD-SGN architecture performs well in terms of the SDN-enable costs and network performance. A good line of future works is to extend the centralized control to computing and memory resources for providing a unified framework of software-defined heterogeneous resources.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

FL designed the SD-SGN architecture; JS designed and carried out experiments; LZ assisted with simulation experiments; DS analyzed experimental results.

FUNDING

This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61231013, 91738301, 91438206, and 91638301), and Program for New Century Excellent Talents in University (Grant No. NCET-09-0025), and Fundamental Research Funds for the Central Universities.

REFERENCES

Amin, R., Reisslein, M., and Shah, N. (2018). Hybrid Sdn Networks: A Survey of Existing Approaches. *IEEE Commun. Surv. Tutorials.* 20 (4), 3259–3306. doi:10.1109/comst.2018.2837161

Arcuri, A. (2019). Restful Api Automated Test Case Generation With Evomaster. *ACM Trans. Softw. Eng. Methodol.* 28 (1), 1–37. doi:10.1145/3293455

Atm, S., Hu, Y-F., Pillai, P., Xu, K., and Baddoo, J. (2017). Optimal Datalink Selection for Future Aeronautical Telecommunication Networks. *IEEE Trans. Aerospace Electron. Syst.* 53 (5), 2502–2515. doi:10.1109/taes.2017.2693227

Bertaux, L., Medjiah, S., Berthou, P., Abdellatif, S., Hakiri, A., Gelard, P., et al. (2015). Software Defined Networking and Virtualization for Broadband Satellite Networks. *IEEE Commun. Mag.* 53 (3), 54–60. doi:10.1109/mcom.2015.7060482

Blial, O., Ben Mamoun, M., and Benaini, R. (2016). An Overview on Sdn Architectures With Multiple Controllers. *J. Computer Networks Commun.* 2016, 1–8. doi:10.1155/2016/9396525

Bohuslava, J., and Martin, J. (2017). "Tcp/ip Protocol Utilisation in Process of Dynamic Control of Robotic Cell According Industry 4.0 Concept," in 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI) (IEEE), 000217–000222. doi:10.1109/sami.2017.7880306

Chen, Z., Qian, H., Mao, Z., Chung, H-M., and Maharjan, S. (2019). "A Study on the Characteristics of Douyin Short Videos and Implications for Edge Caching," in Proceedings of the ACM Turing Celebration Conference-China, 1–6. doi:10.1145/3321408.3323082

Chowdhury, T., and Alam, M. J. (2019). Performance Evaluation of Tcp Vegas Over Tcp Reno and Tcp Newreno over Tcp reno. *JOIV: Int. J. Inform. Visualization.* 3 (3), 275–282. doi:10.30630/joiv.3.3.270

Cisco Annual Internet Report (2021). Cisco Annual Internet Report (2018–2023) White Paper Website. Available at: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

Do, S., Luong, V. e., Paul Lin, B. S., and Tung, L-P. (2019). Sdn/nfv Based Internet of Things for Multi-Tenant Networks. *Trans. Networks Commun.* 6 (6), 40. doi:10.14738/tnc.66.5695

Du, L., Steenkamp, T., Kaplan, S., and Wilkinson, R. H. (2009). Wireless Sensor Network Gateway. *AFRICON.* 2009, 1–6. doi:10.1109/afrcon.2009.5308112

Giovane, C. M., Heidemann, J., Schmidt, R. d. O., and Hardaker, W. (2019). "Cache Me if You Can: Effects of Dns Time-To-Live," in Proceedings of the Internet Measurement Conference, 101–115.

Haveliwala, S., Vyas, D., and Shah, H. (2020). Dynamic Congestion Control Mechanism in Mobile Adhoc Network: Tcp Westwood-dcc. *Ijrte.* 8 (5), 4784–4789. doi:10.35940/ijrte.e6828.018520

Hilvert-Bruce, Z., Neill, J. T., Sjöblom, M., and Hamari, J. (2018). Social Motivations of Live-Streaming Viewer Engagement on Twitch. *Comput. Hum. Behav.* 84, 58–67. doi:10.1016/j.chb.2018.02.013

Hoogendoorn, I. (2021). Nsx-t Nat, Dhcp, and Dns Services. *Multi-Site Netw. Security Serv. NSX-T.*, 87–124. doi:10.1007/978-1-4842-7083-7_4

Kawaguchi, E., Kasuga, H., and Shinomiya, N. (2019). "Unsplittable Flow Edge Load Factor Balancing in Sdn Using P4 Runtime," in 2019 29th International Telecommunication Networks and Applications Conference (ITNAC) (IEEE), 1–6. doi:10.1109/itnac46935.2019.9077984

Li, K., Guo, B., Huang, S., Guo, X., Dong, T., Yin, J., Liu, Z., Zhang, T., and Du, J. (2019). "A Simulation Platform for Software Defined Integrated Space Ground Network," in 2019 IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT) (IEEE), 118–123. doi:10.1109/icait.2019.8935928

Lin, P., Bi, J., and Hu, H. (2016). Btsdn: Bgp-Based Transition for the Existing Networks to Sdn. *Wireless Pers Commun.* 86 (4), 1829–1843. doi:10.1007/s11277-015-3145-0

Miraz, M. H., Molvi, S. A., Ganie, M. A., Ali, M., and AbdelRahman, H. (2017). Simulation and Analysis of Quality of Service (Qos) Parameters of Voice Over Ip (VoIP) Traffic Through Heterogeneous Networks. *Int. J. Adv. Comput. Sci. Appl.* 8 (7), 242–248.

McDowell, J. C. (2020). The Low Earth Orbit Satellite Population and Impacts of the Spacex Starlink Constellation. *ApJ.* 892 (2), L36. doi:10.3847/2041-8213/ab8016

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., et al. (2008). OpenFlow. *SIGCOMM Comput. Commun. Rev.* 38 (2), 69–74. doi:10.1145/1355734.1355746

Miano, S., Risso, F., and Woesner, H. (2017). "Partial Offloading of Openflow Rules on a Traditional Hardware Switch Asic," in 2017 IEEE Conference on Network Softwarization (NetSoft) (IEEE), 1–9. doi:10.1109/netsoft.2017.8004107

Phemius, K., Bouet, M., and Leguay, J. (2014). "Disco: Distributed Multi-Domain Sdn Controllers," in 2014 IEEE Network Operations and Management Symposium (NOMS) (IEEE), 1–4. doi:10.1109/noms.2014.6838330

Popovski, P., Trillingsgaard, K. F., Simeone, O., and Durisi, G. (2018). 5g Wireless Network Slicing for Embb, Urllc, and Mmtc: A Communication-Theoretic View. *IEEE Access.* 6, 55765–55779. doi:10.1109/access.2018.2872781

Radivilova, T., Kirichenko, L., Ageyev, D., Tawalbeh, M., and Bulakh, V. (2018). "Decrypting Ssl/tls Traffic for Hidden Threats Detection," in 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT) (IEEE), 143–146. doi:10.1109/dessert.2018.8409116

Roy, A., Acharya, T., and DasBit, S. (2018). Quality of Service in Delay Tolerant Networks: A Survey. *Computer Networks.* 130, 121–133. doi:10.1016/j.comnet.2017.11.010

Saito, T., Tomoda, I., Takabatake, Y., Arni, J., and Teramoto, K. (2000). Home Gateway Architecture and its Implementation. *IEEE Trans. Consumer Electron.* 46 (4), 1161–1166. doi:10.1109/30.920474

Santos, L., Gómez, A., and Sarmiento, R. (2019). Implementation of Ccsds Standards for Lossless Multispectral and Hyperspectral Satellite Image Compression. *IEEE Trans. Aerospace Electron. Syst.* 56 (2), 1120–1138. doi:10.1109/TAES.2019.2929971

Šeremet, I., and Čaušević, S. (2020). "Advances of Configuring Quality of Service (Qos) in Software Defined Networks (Sdn) by Using Meter Table," in 2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH) (IEEE), 1–5. doi:10.1109/infoteh48170.2020.9066275

Uddin, M., Mukherjee, S., Chang, H., and Lakshman, T. V. (2018). Sdn-Based Multi-Protocol Edge Switching for Iot Service Automation. *IEEE J. Select. Areas Commun.* 36 (12), 2775–2786. doi:10.1109/jsac.2018.2871325

Wallshein, C. (1999). *Tactical Data Link Gateways Introductiontactical Data Link Gateways Introduction* Washington, DC: Air Force Studies and Analyses Agency.

Wang, Q., Zhu, X., Ni, Y., Gu, L., and Zhu, H. (2020). Blockchain for the Iot and Industrial Iot: A Review. *Internet of Things.* 10, 100081. doi:10.1016/j.iot.2019.100081

Zhu, Q., Wang, R., Chen, Qi., Liu, Y., and Qin, W. (2010). "Iot Gateway: Bridgingwireless Sensor Networks into Internet of Things," in 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (IEEE), 347–352. doi:10.1109/euc.2010.58

Zhu, X., Jiang, C., Kuang, L., Ge, N., and Lu, J. (2017). Non-Orthogonal Multiple Access Based Integrated Terrestrial-Satellite Networks. *IEEE J. Select. Areas Commun.* 35 (10), 2253–2267. doi:10.1109/jsac.2017.2724478

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.