Big Data Orchestration as a Service Network

Xiao Liu, Yuxin Liu, Houbing Song, and Anfeng Liu

The authors argue that a big data network joint SDN, together with cloud and fog computing platforms, can build a service chain network. In SDN, the purpose is to reduce a large amount of redundant data and response time. They propose a novel Big Data Orchestration as a Service Networking framework, which can dynamically orchestrate big data into services in SDN.

Abstract

This article argues that a big data network joint SDN, together with cloud and fog computing platforms, can build a service chain network. In SDN, the purpose is to reduce a large amount of redundant data and response time. We propose a novel Big Data Orchestration as a Service (BDOaaS) as the networking framework, which can dynamically orchestrate big data into services in SDN. In BDOaaS networking, the data center distributes software to all devices in the distributed network, which can orchestrate big data into services in the distributed network; the services-oriented network model is formed. Thus, the network load and response time is reduced. The BDOaaS framework and various components of BDOaaS as well as operation mechanisms are discussed in detail. Simulation results are presented to show the effectiveness of the proposed BDOaaS framework. In addition, we discuss a number of challenges in implementing the proposed framework in next generation networks.

Introduction

Nowadays, big data networks and software defined networks (SDNs) are booming [1–3]. The development of big data networks is based on the enhanced ability to collect and process data, which can make human decision making more comprehensive and wise [1, 3]. Big data was developed due to the development of new data acquisition equipment:

1. Pervasive mobile devices such as smartphones are expected to play a significant role, for example, as human-machine interfaces, data sources for environmental monitoring (i.e., sensors for context detection) and user profiling, ubiquitous social networking, and anywhere-anytime-to-anything connectivity [2–5].
2. Sensing devices will be improved. For example, in a manufacturing plant, ubiquitous sensors gather monitor data to support the work [3, 6, 7].

This unprecedented increase in data traffic faces unprecedented challenges for current levels of data transmission:

1. The amount of data cause larger network loads. In big cities, many forms of data are relevant; thus, it is important to propose a method to reduce the amount of data.
2. Due to the longer distance between network edges and core networks, when users request services, the request information is transmitted to the core network, and the service is satisfied.

This requires more time and large delays. Thus, the required time for each service must be reduced. This can cause network congestion and rapid increases in delay, and quality of service (QoS) deteriorates rapidly; thus, the quality of experience (QoE) is worse [8].

In big data networks, there are three different roles. (a) service provider (SP), (b) services customer (SC) or simple user, and (c) big data collector (BDC), which refers to devices or people that can collect data. The aim of the SP is to combine the collected data into advanced services and send services to users after cleaning and refining the data. Usually, the SP is the owner of the data center and distributes the data collection task to the BDC. The BDC collects data through sensing devices and then transmits those data packets to the data center. The SP provides services to users after processing the data. For example, VTrack, which provides omnipresent traffic information, and NoiseTube, which makes noise maps, can be regarded as SPs [7]. Crowd sensing networks (CSNs), which leverage the ubiquity of sensor-equipped mobile devices, are a kind of BDC that can collect information and provide a new paradigm for solving complex sensing applications [2, 9]. For example, users can obtain traffic information or noise map services from VTrack or NoiseTube, respectively. In this application, the SP needs to collect a large amount of data and then provide users with detailed traffic information or noise distribution. In order to improve QoS, it is better to collect a large amount of data. However, a large amount of data can increase the network load.

The SP usually adopts a data center or an information-centric (IC) system, which is located in the cloud, to store data [2, 10]. IC systems and data centers have been extensively studied in recent years [10]. All data need to be sent to the data center, and the BDC for collecting data is at the edge of the network. Because the distance between the cloud and the edge device is usually long, transmitting a large amount of data from the device to the SP may not be feasible or economical. If users are at the edge of the network, the SP may not provide guaranteed low-latency service to users. Thus, the establishment of an effective big data network faces huge challenges.

To address these issues, content-based distribution technology is proposed. In-network caching is used in an information-centric network (ICN) to speed up content distribution and improve net-

Digital Object Identifier:
10.1109/MCOM.2017.1700090

Xiao Liu,Yuxin Liu, and Anfeng Liu are with Central South University; Houbing Song is with Embry-Riddle Aeronautical University.

work resource utilization [2, 10]. In ICNs, requests no longer need to travel to the content source but are served by a closer ICN "content node" along the path, which can speed up the distribution of content [10]. The shortages of this scheme are that it is a kind of improvement strategy and thus will not fundamentally solve the problem because the proportion of caching content on the network is not high. In addition, this method is only suitable for data requests and cannot effectively alleviate the generated network load that the primary data flows upload to the data center.

The main reason for low network performance is the long distance route between BDCs at the network edge and the data center. Fog computing and mobile edge computing (MEC) [5] have been proposed to deploy local data centers closer to end users. Fog (from core to edge) computing, a term coined by Cisco in 2012, is a distributed computing paradigm that empowers network devices at different hierarchical levels with various degrees of computational and storage capability [5]. Compared to cloud computing architecture, which is centralized in nature [5], the scheme provides real-time and low-latency services to billions of IoT devices at the edge of the network [5]. Due to the increase in the fog layer in the system model, a large amount of data and service requests can be processed in the local network but do not need be sent to the data center in the cloud. This layered structure can improve network performance to a certain extent, while this improved performance is achieved by hardware, thus increasing the cost of the SP and decreasing flexibility. Moreover, the local center in the network edge needs to interact with other data centers, which also creates a network load. Therefore, an advanced network is needed that can reduce the network load, making the network more extensible and providing better QoE and QoS.

SDNs have attracted great interest in both academia and industry [2, 11]. SDNs introduce the ability to program the network via a logically software-defined controller. SDNs can be extended to wireless networks, which benefit from the SDN [2, 11].

The rise of SDNs has brought opportunities to solve these problems. This article argues that big data networking jointly with SDN, together with cloud and fog computing platforms, can build a service chain network. Through networks such as ICNs, service requests are processed in less time; caching the same packets at the appropriate time and path is not an easy thing. We propose a novel BDOaaS networking framework that can enable dynamic orchestration of big data into services in SDN to meet the requirements of next generation networks. In BDOaaS, the data center distributes software to all levels of devices in the distributed network, which can orchestrate big data into services in all devices of a distributed network, not only in the data center, thus achieving a services-based network computing model. Orchestrating data as a service is achieved by software stored at all levels of devices in the network. This kind of software is the software for data collection tasks published by the SP, which plays a role in two areas.

Data aggregation. When the BDC submits the data flow to the data center, all devices that pass by call the software for operation data aggregation [3, 7], which can reduce the amount of uplink data flow greatly, thus reducing the network load.

Services build. The data are not only aggregated when the data flow is transmitted to devices but also computed to build the service. Thus, when users request a service, if the services structured by current devices can meet the requests, the results will be immediately returned to the users. Thus, service request flow and services return flow are reduced. Because different devices are in the different network layers, the structured services are different. Devices on the network edge only construct services in the scope of a local small area, and devices near the core network construct services in the greater range of a region. Thus, devices at a lower level can satisfy most service requests by users. Therefore, the system has strong adaptability and expansibility.

Devices on the network edge only construct services in the scope of a local small area, and devices near the core network construct services in the greater range of a region. Thus, devices at a lower level can satisfy most service requests by users. Therefore, the system has strong adaptability and expansibility.

In this article, the proposed BDOaaS has the following characteristics:
- The main target is to orchestrate service; it is different from current networks whose target is data transmission. Moreover, distributed orchestration is operated in all levels of network equipment, and its target is to build a kind of service-based distributed network.
- Data can be transmitted after data has been programmed; thus, the amount of data can be reduced greatly.
- An SP releases software to network equipment, and data services are transformed in the distributed network, thereby greatly reducing the amount of data.

Compared to traditional networks, the main innovations of BDOaaS are as follows.

•A new services-based computing network model is introduced that can run in the cloud and fog computing platform, extending the functionality of big data networks and SDN. In a services-based computing model, the raw data are no longer routed in the network. The two most important orchestration functions are as follows:

–Aggregating data reduce data flow.

–The data are programmed as services.

BDOaaS combines data and software to orchestrate services.

•A new way to reduce network load is conceived. The core idea is that it routes services rather than data. In the previous network, the main function of the network is to receive and forward data, but not take into consideration data content. On the other hand, the BDOaaS can be regarded as a content-based (services-based) network, in which the SP distributes the software to all levels of devices, and then data are orchestrated into services to reduce the network load.

•The BDOaaS networking framework is proposed. The components of the BDOaaS framework are discussed in detail, which has been proven effective by experiments.

However, there are many challenges for providing a better method. For example:
- Methods that are compatible with the current network
- The design of orchestrating software
- Releasing orchestrating software

A detailed discussion is presented in the section on open research challenges.

The BDOaas network framework is useful for networks of large size. For example, for big data networks, the amount of raw data is large. For large cities, the data center can collect thousands of raw data packets each minute.

The rest of this article is organized as follows. We describe an overview of the BDOaaS networking framework. We present the operating mechanism in BDOaaS. Simulation results are presented. Some open research issues are discussed. Finally, we conclude this work.

Overview of Orchestrating Data as Services Networking

Motivations

In the last decade, there has been a rise in a variety of ubiquitous computing and handheld devices [2, 4, 6, 7]. Large amounts of data greatly enhance the human ability to observe the world and make wise decisions; a network with a large amount of data transmission is called a big data network. In the big data network, a large amount of data needs to be transmitted; there are more than 9 billion devices used to generate data according to statistics, with an annual growth rate of 50 percent [5]; the rate of increase of the trunk network is <10 percent [5]. Thus, it becomes an urgent task to establish a new network computing model to cater to the rapid growth in the big data network.

BDOaaS is proposed based on two aspects.

1. A more advanced services-based network framework is needed for the development of big-data-based applications. We note that a large amount of forwarding data in the big data network are relevant; the relevant data always contain the same information. Thus, only some of that information is transmitted to the next devices, and devices can know all the information. Thus, some redundant data can be refined through the appropriate calculation; this operation is called aggregating data (or orchestrating data). For example, a meteorological department (as an SP) releases tasks for gathering haze information in the specified area, and a huge population with smart devices or devices collects the haze information and then submits the haze information to the SP (the SP's data center). In such applications, there are a large number of data packets, which can cause greater network load if all packets are routed to the data center. We note that the content of raw packet is {time, location, content}, and the sensed haze information in small areas is the same at the same time; thus, the data packet can be orchestrated as a new data packet by network devices through the orchestrating software provided by SP. The most simple form is {time, location1, location2,... location*n*, content}, and the amount of data can be reduced comparatively. If the haze information in all locations is the same, the haze information can be expressed as a data packet, such as {time, all-location, content}. Thus, the amount of data can be reduced by an order of magnitude. However, orchestrating software is distributed by specific software released by the SP. The difference with the previous network is that orchestrating software not only resides in network devices but also directly resides in BDC devices; thus, orchestrating data is more effective. For example, if weather conditions or industrial monitoring situations are stable, and the sensed object (e.g., haze information) has not changed for a long time, its report interval can be increased, and it will not have any impact on the application. On the contrary, if the sensed object changes dynamically, the ample data frequency can be increased adaptively, which can be effective to ensure the quality of data acquisition and greatly reduce the network load.

From the viewpoint of building and obtaining services, information based on big data, such as VTrack, NoisTub, and Haze information, due to orchestrating software, resides in the devices of the uplink flow of raw data. The query information of users has already been built; thus, users can send a services request query; the format of Query haze information services, for example, is {time, location, request-content}. If the information requested by users can be satisfied by devices in the routing path, the devices can directly return the results, without routing to the data center. Even if devices cannot satisfy the request, the request information continues routing to a higher level of devices. Because most users' service requests are local, most requests could be returned in the network edge. As a result, the network load and the service request time are reduced, and the user QoE is improved.

2. From the point of view of hardware facilities in the current network, in big data networks, fog computing combined with SDN has the corresponding conditions to implement BDOaaS. This is the case in particular for the fog computing platform, the purpose of which is to move network computing to the network edge; thus, the method is similar to this scheme, where BDOaaS moves the network to the network edge. Thus, the establishment of BDOaaS in the fog computing platform is a natural thing. However, the fog computing platform provides a physical platform, and the implementation aspect is solved by BDOaaS. Therefore, BDOaaS has good significance.

The BDOaas network framework is useful for networks of large size. For example, for big data networks, the amount of raw data is large. For large cities, the data center can collect thousands of raw data packets each minute. If every raw data packet is transmitted to the data center, the network load is 100 MB in a round data collection; the generated traffic flow within one hour is up to 6 GB; however, the capacity of final distribution of the haze map is only a few kilobytes for collecting data in a 10,000-point location in one hour. If the BDOaaS is used, data packets will be merged into one packet in one grid, and the data can be merged among different grids again; the amount of orchestrating data transferred is less than 60 MB in 1 hour, and thus the data transmission can be reduced by a factor of 100.

The Framework of BDOaaS

The framework of BDOaaS is shown as Fig. 1. BDOaaS can be divided into five layers: the data collection layer, fog or edge network layer, core network layer, data center layer, and application layer.

Data collection layer: This is located at the edge of the network and consists of pervasive sensing devices, such as various types of mobile devices, smartphones, and industrial, civil, and public area sensing devices [12, 13]. All devices for collecting data are called BDCs. These devices are the data source of BDOaaS. According to [5], the number of connected devices has exceeded the number of people on Earth since 2011. Connected devices now number 9 billion and are expected to grow more rapidly, reaching 24 billion devices by 2020 [5]. With the increasing number of heterogeneous devices connected to IoT generating large amounts of data [5], using a data-transmission-based network cannot meet the current rapid growth of data. In BDOaaS, the raw data are orchestrated and then form services; thus, the data transmission flow is reduced greatly to meet the development of the network.

Fog or edge network layer: However, in the BDOaaS, those network devices in the fog layer are equipped with the software released by SP and orchestrate data as services. Thus, the devices in the network layer have different functions compared with the previous devices.

•If those devices receive the data that need to be uplink forwarded, those data can be orchestrated by the interface provided by the SP.

•If they receive the service request, they call the interface provided by the SP and then orchestrate those data. If the orchestrated data can meet user requests, the results can be returned to the users; otherwise, the request continues forwarding to the uplink.

•If those devices receive a services return, they not only forward services but also integrate the services and the owner services; thus, the service requests of downlink users can be satisfied locally. Network devices orchestrate each uplink data packet; thus, the local services can be updated, and the downlink services can be orchestrated. Thus, the data that those services include come from larger areas. For example, in NoiseTube applications, the local updated noise distribution may be obtained after network devices orchestrate the received uplink data. Meanwhile, the noise distribution in a wider range can be obtained after the downlink services are orchestrated, which can provide a noise query service in a short path style to help reduce the service request delay and improve user QoE.

Core network: The core network refers to the current backbone network. Its function is similar to the devices in the fog layer. However, these devices are located in the upper layer, which has a larger view and can provide more functionality.

Data center layer: This is large-scale equipment with huge memory capacity and computing ability disposed by the SP; it can analyze and process data in depth. However, the difference from the previous network is that the data center in the previous network is mainly used to store and process data. In the BDOaaS, the SP not only has large-scale software, which processes data locally, but also provides software for orchestrating data to network devices. This software is programmed for specific applications. When the SP publishes the data collection task, it releases specific software to the network devices. Thus, in BDOaaS, the data center not only has a function for storing and processing data but also for strutting and releasing software, so the BDOaaS forms a service-oriented network that combines data and SDN.

Application layer: This refers to users for the application of services.

These five components in different situations have multiple roles. In the data collection layer,

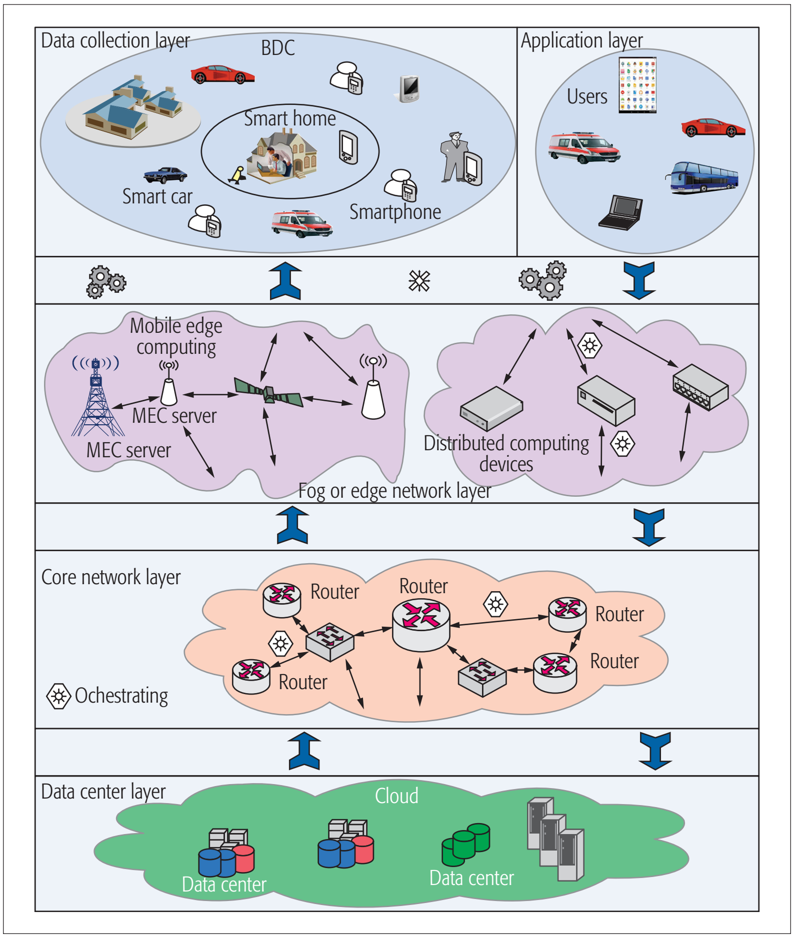

Figure 1. The framework of BDOaaS.

a smartphone can be used as a data collection device (i.e., a BDC). If devices can request services, they can be called users. In addition, there is no clear boundary between the fog layer and the core network layer. In addition, some big data centers in the network edge can be regarded as data centers. Figure 1 shows a typical network structure diagram. Compared to the core network, the fog layer mainly relies on distributed computer resources near local devices. Operations regarding data, data processing, and applications in the fog layer are more dependent on local devices rather than servers. Unlike the core network, almost all devices are stored in the core network.

The Running Mechanism in BDOaaS

There are four kinds of flows in BDOaaS. These four types of flow are as follows.

Data or services flow: The content of the route is the data collected by the BDC, and the flow direction is from devices on the network edge to the data center. In BDOaaS, because network devices in the routing path orchestrate data, data are transformed into services. The transformed services will be transmitted to the data center; this is called the data or services uplink route.

Services request flow: The user sends a request to the SP to ask for a service. The flow direction is from the user to the SP (i.e., the data center); this is called the service uplink route.

Service return flow: After the SP or network

The main idea of BDOaaS is that the SP releases program codes for orchestrating data to all network devices, and then refines the forwarding data to reduce the amount of data. Its goals are to reduce the network load and enhance user QoS and QoE.

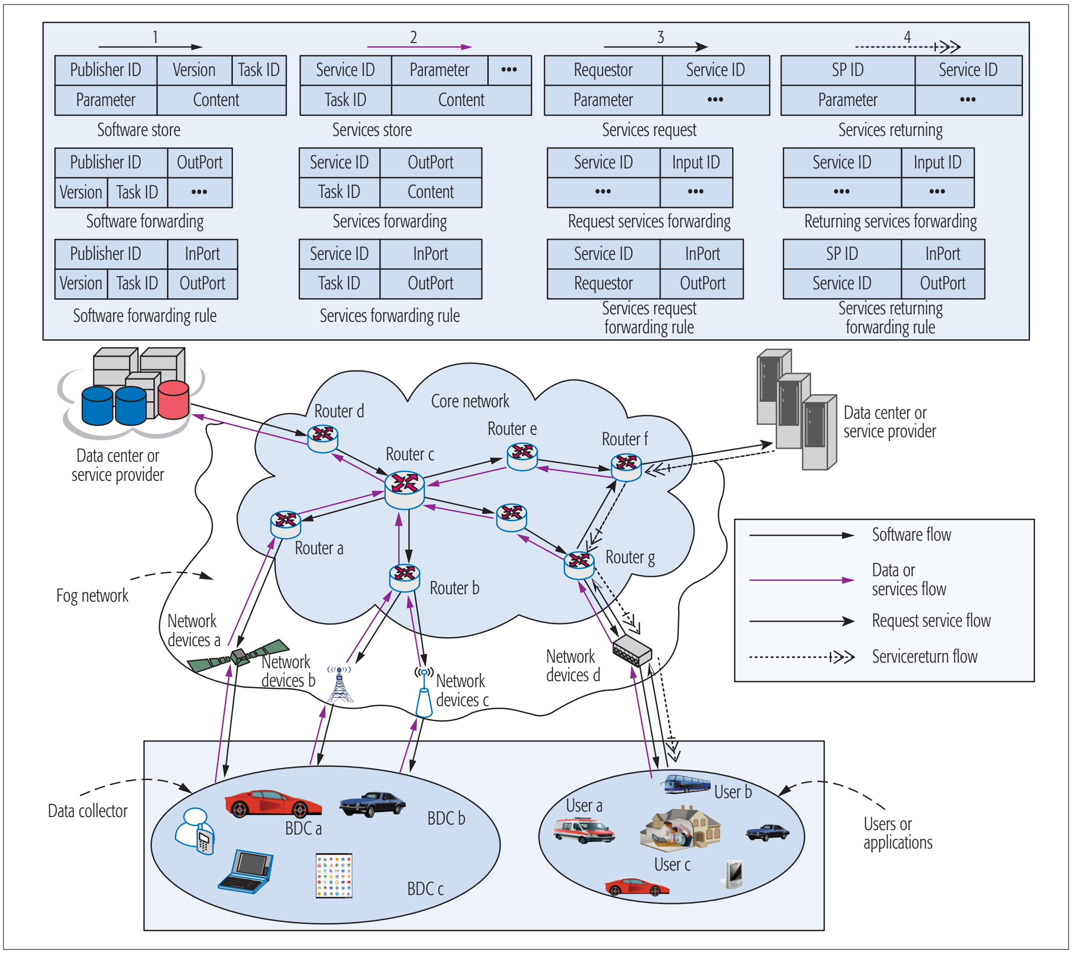

Figure 2. The running mechanism in BDOaaS.

devices receive a service request from users, if the services orchestrated by devices can meet the request of a user (the SP must be able to meet the user request), the device will send the service flow to the user, which is called the service downlink route.

Software release flow. The network devices can work effectively only when the devices receive software orchestrated by the SP; thus, the SP needs to release the software to the network devices, referred to as software flow.

The running mechanism in BDOaaS can be illustrated in Fig. 2. These four flows can show the operating mechanism.

1. First, a BDC obtains data by sensing and measuring the surrounding environment and then submits those data to the network; the operation is shown in flow 2 of Fig. 2 (data or services flow). BDC a submits its sensing data to network device b, and network device b receives the collected data from other BDCs at the same time. Thus, network device b orchestrates data through software provided by the SP. The orchestrated data may be services or aggregated data. Then the data or service flow continues flowing to the data center; this route is routed through router b, router c, router e, and router f. Each network device can orchestrate the data or services in the routing path; the route data are refined after devices orchestrate the data once. Finally, the data center obtains all information.

2. Services request flow. This is shown as Fig. 2; user a requests a service from the SP. After network device d receives the request, it checks whether the service request can be satisfied; if it can be satisfied, device b returns a service result or continues forwarding the service request to the SP. All devices (e.g., router g, router f) in the routing path check whether the service meets the request; if the request service can be satisfied, the service result can be returned, or the service request continues to be forwarded. If no devices in the routing path can satisfy the service request, the SP must be able to meet the service request of the user.

3. Services return flow. After network devices satisfy the service request of the user, the service results can be returned along the original routing path.

4. Software flow. The SP (residing in the data center) customizes software for orchestrating data to services. The customized software is spread to devices in the designated area (including the wireless network and the BDC) to complement BDOaaS.

The main idea of BDOaaS is that the SP releases program codes for orchestrating data to all network devices, and then refines the forwarding data to reduce the amount of data. Its goals are to reduce the network load and enhance user QoS and QoE. First, the SP releases the program code for orchestrating data to all levels of network devices (Fig. 2) for special functions, and all devices receive software. In the process of uploading

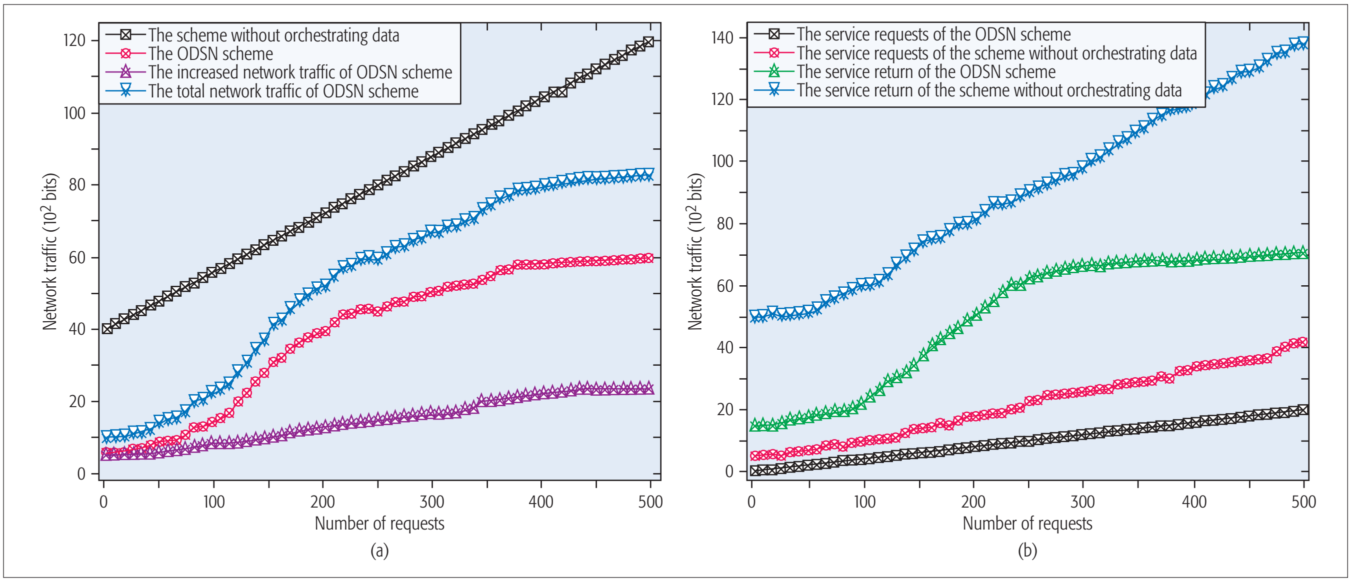

Figure 3. The data traffic of BDOaaS.

data from the BDC to the SP, when all devices receive data from downstream devices, the received data can be fused (or the services can be re-orchestrated) according to received software; the amount of data can be reduced greatly while ensuring complete information. Then the fused data continue to be transmitted to the data center. When users request services, if devices near users can orchestrate this service according to the received software, it returns the service to the users. Thus, the delay can be reduced.

Simulation Results

The main goals of this article are:
- Reduce the amount of redundant data.
- Speed up the response of the service request.

The model for orchestrating sensed data is as follows. If current device A receives the raw data p_a and receives the raw data p_b from sensing devices, the results for aggregating data are $\xi(A, B) = \max(p_a, p_b) + (1 - \beta_{A,B})\min(p_a, p_b)$, where

$$\beta_{A,B} = \frac{1}{e^{(d_{A,B}^2/\alpha)}}, d_{A,B}^2$$

is the distance between devices A and B, and α is a constant coefficient. If device A receives the aggregated data δ_b from device B, as long as one data packet in the received packet is not raw data, the result for aggregating data is $\xi(A, B) = \max(p_a, p_b) + (1 - \beta_{A,B})\min(p_a, p_b)$, where η is the forgetting factor, a decimal number less than 1. The parameters are p_a = 30bits, $\eta = 0.8$, α =2.

In the simulations, we consider one eNodeB, one WiFi router, and one WiMAX router, each of which has 10 users in the network. The kind of requirement service of users is 100. Assume that there are 100 popular content requests, and there are a total of 1000 requests for popular content, caching, and computing from all 30 users randomly. Assume the transmission delay is a value in the range of [10, 20] ms per hop. The storage space of devices is 500 bits. We consider the average response delay of users' requests and the average network traffic as our performance metrics.

The traffic generated with the number of requests is shown in Fig. 3. If the method for orchestrating data is not used, the generated traffic is given in the black line of Fig. 3. It can be seen that the generated traffic is basically linear growth with the number of requests. However, if the method for orchestrating data is adopted, the data may be uploaded after a large amount of data is orchestrated, so the growth rate is moderate. However, in BDOaaS, due to the release software, the increased flow for releasing software is given in Fig. 3a. In BDOaaS, the total traffic is shown in the blue line of Fig. 3a. A comparison of generated traffic for service requests is given in Fig. 3b. The generated traffic for service requests is small, and for return services, it is big. This shows the effectiveness of the BDOaaS.

Figure 4 shows the average number of hops to complete a full service and delay.
- Comparison of services request hops. Obviously, in BDOaaS, most services requests will be satisfied at the edge of the network; thus, the hop number in BDOaaS is far lower than that of other networks.
- Comparison of delay. Delay is related to network hop count and network congestion. If the hop count is large, the delay is higher. Obviously, the hop count in BDOaaS is smaller than the hop counts in other networks. Thus, the delay in BDOaaS is small.

Open Research Challenges

Despite the potential of BDOaaS networking, there are some research challenges that must be addressed. In this section, we address some of these challenges, which help to promote the study of BDOaaS.

Compatibility

The most important issue is compatibility for BDOaaS. In the proposed BDOaaS, we assume that each network device can receive software released by the SP, and can orchestrate data and services. However, designing current network devices should be as simple as possible. In particular, for the backbone of the high-speed network, complex data processing and comput-

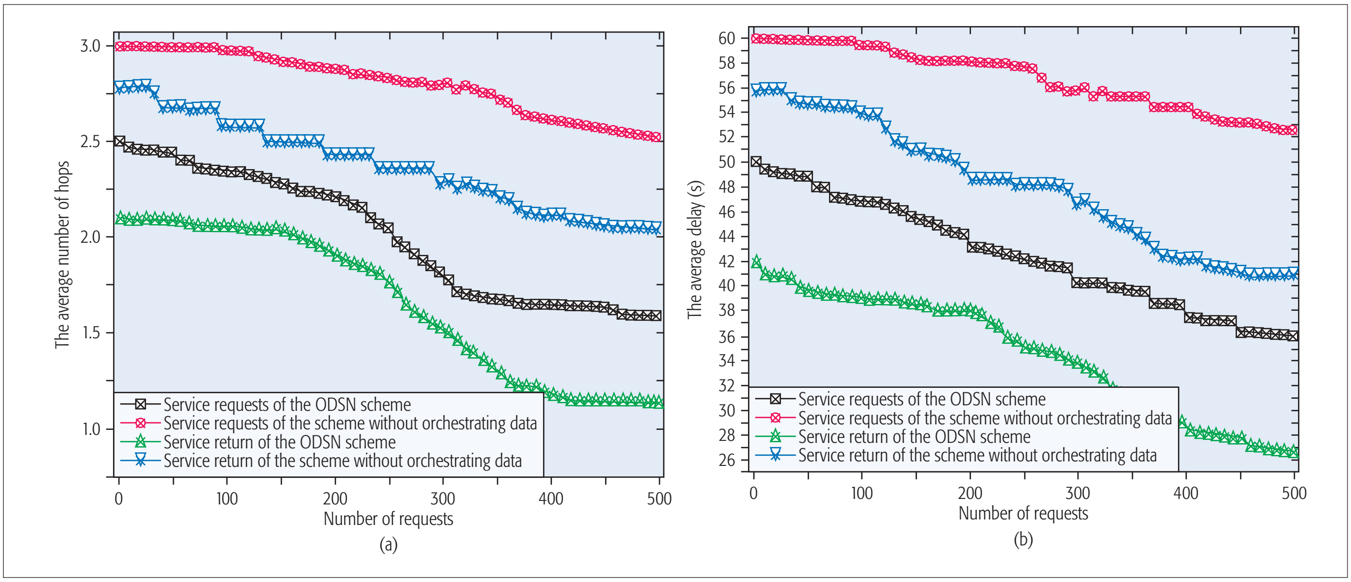

Figure 4. The quality of service of BDOaaS.

ing will greatly reduce the speed of forwarding data. Thus, the compatibility of BDOaaS with the current network is an important issue. In the current network, first, some specialized devices for orchestrating services can be deployed in the network edge (also in the core network), so BDOaaS can be achieved to some extent. Second, the network uses a unified format; the devices can process data packets quickly. Third, the data have been pre-processed by source collector devices. Due to the large number of devices of the BDC, the pre-processed function on the BDC devices will not reduce network performance but can effectively reduce the required computing for orchestrating data in network devices. With the development of hardware for network devices, the enhanced ability for computing and processing data will help the BDOaaS become more achievable.

The Issue of Orchestrating Software

The design of the orchestrating software is another important challenge. The orchestrating software in BDOaaS requires the following characteristics:

1. Small size. Software can quickly spread to all levels of devices in the network and lead to small network loads. Small-sized software can be stored in network devices with limited storage space.

2. The calculation is simple, and the speed of calculation is fast, to make it possible to adapt to the network devices with high-speed forwarding data. Meanwhile, upgrading old software is important in the BDOaaS.

The Routing Mechanism for Services

The current network is based on IP routing. Although there is a routing strategy based on content [14], there are still many issues worth studying. In the BDOaaS, the best routing method is based on services routing; this method is regarded as special routing based on content. Because each service has a service ID, it is more complex to design routing based on services than on the current IP. In the BDOaaS, releasing the orchestrating software is another important issue. Such software usually needs to spread to devices in a specific geographic area. The previous broadcast and multicast mode are not suitable for a route whose sensed target area is a geographic region. For routing that releases software to mobile devices, the design faces challenges. How to detect and update software is another important issue. Due to the large number of devices, the version number of software received by multiple devices is different; thus, how to ensure consistency among different software versions is a new issue.

In fact, the BDOaaS, as a new network, faces many challenges. Security is a permanent theme in the network and is even more important in the BDOaaS. Most services data based on big data are from the data collected by smartphones, and there are therefore greater threats to people's privacy, security, and so on [13]. In the network, designing incentive mechanisms for collecting data in addition to the allocation mechanisms of the network source face challenges in the BDOaaS.

Conclusion

In the future, the ability to obtain data will grow rapidly, and the growth of network transmission capacity will be slow; the contradiction between the two will become increasingly prominent. We believe that a fundamental solution is to construct a network that transmits information after refining data in the future network [15]. In fact, networks that transmit information instead of raw data have a variety of implementations and face many challenges. In this article, we propose a novel BDOaaS networking framework that can dynamically orchestrate big data to services in SDN, reducing the amount of data and network delay and thus improving user QoE. In BDOaaS, orchestrating data as services or orchestrating services as services is a novel idea for constructing future networks.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61379110, 61073104, 61572526, 61272494, 61572528) and the National Basic Research Program of China (973 Program) (2014CB046305)

References

[1] E. Zeydan *et al.*, "Big Data Caching for Networking: Moving from Cloud to Edge," *IEEE Commun. Mag.*, vol. 54, no. 9, Sept. 2016, pp. 36–42.

[2] R. Huo *et al.*, "Software Defined Networking, Caching, and Computing for Green Wireless Networks," *IEEE Commun. Mag.*, vol. 54 no. 11, Nov. 2016, pp. 185–93.

[3] L. Kong *et al.*, "Embracing Big Data With Compressive Sensing: A Green Approach in Industrial Wireless Networks," *IEEE Commun. Mag.*, vol. 54, no. 10, Oct. 2016, pp. 53–59.

[4] Y. Liu *et al.*, "APMD: A Fast Data Transmission Protocol with Reliability Guarantee for Pervasive Sensing Data Communication," *Pervasive and Mobile Computing*, DOI:10.1016/j.pmcj.2017.03.012, 2017.

[5] S. Sarkar and S. Misra, "Theoretical Modelling of Fog Computing: A Green Computing Paradigm to Support IoT Applications," *IET Networks*, vol. 5, no. 2, 2016, pp. 23–29.

[5] A. Liu *et al.*, "Distributed Multi-representative Re-Fusion Approach for Heterogeneous Sensing Data Collection," *ACM Trans. Embedded Computing Systems*, 2017, vol. 16, no. 3, 73, DOI: http://dx.doi.org/10.1145/2974021.

[6] X. Liu *et al.*, "Large-Scale Programing Code Dissemination for Software Defined Wireless Networks," *Computer Journal*, DOI: 10.1093/comjnl/bxx014, 2017.

[8] H. Li *et al.*, "Engineering Searchable Encryption of Mobile Cloud Networks: When QoE Meets QoP," *IEEE Wireless Commun.*, vol. 22, no. 4, Aug. 2015, pp. 74–80.

[9] J. Wang, C. Hu, and A. Liu, "Comprehensive Optimization of Energy Consumption and Delay Performance for Green Communication in Internet of Things," *Mobile Info. Systems*, vol. 2017, article ID 3206160, DOI: 10.1155/2017/3206160.

[10] C. Fang *et al.*, "A Survey of Green Information-Centric Networking: Research Issues and Challenges," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 3, 2015, pp. 1455–72.

[11] K. Xu *et al.*, "Toward Software Defined Smart Home," *IEEE Commun. Mag.*, vol. 54, no. 5, May 2016, pp. 116–22.

[12] X. Liu *et al.*, "Trace Malicious Source to Guarantee Cyber Security for Mass Monitor Critical Infrastructure," *J. Computer and System Sciences*, 2016, DOI: 10.1016/j.jcss.2016.09.008.

[13] I. M. Butun *et al.*, "Cloud-Centric Multi-Level Authentication as a Service for Secure Public Safety Device Networks," *IEEE Commun. Mag.*, vol. 54, no. 4, Apr. 2016, pp. 47–53.

[14] J. M. Batalla *et al.*, "ID-based Service-Oriented Communications for Unified Access to IoT," *Computers & Electrical Engineering*, vol. 52, 2016, pp. 98–113.

[15] Y. Kryftis *et al.*, "Efficient Entertainment Services Provision over a Novel Network Architecture," *IEEE Wireless Commun.*, vol. 23, no. 1, Feb. 2016, pp. 14–21.

Biographies

Xiao Liu received M.Sc. degrees from Central South University, China, in 2017. Currently she is a Ph.D candidate with the School of Information Science and Engineering of Central South University, China. Her research interests are crowd sensing networks, wireless sensor networks, and wireless security.

Yuxin Liu is currently a student in the School of Information Science and Engineering of Central South University. Her research interests are services-based networks, crowd sensing networks, and wireless sensor networks

Houbing Song [M'12, SM'14] (Houbing.Song@erau.edu) received his Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, in 2012. In 2017, he joined the Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, Florida, where he is currently an assistant professor and the founding director of the Security and Optimization for Networked Globe Laboratory (SONG Lab, www.SONGLab.us). He is an Associate Technical Editor of *IEEE Communications Magazine*. He is a Senior Member of ACM. His research interests lie in the areas of communications and networking, cyber-physical systems, cybersecurity, and big data analytics.

Anfeng Liu received his M.Sc. and Ph.D degrees from Central South University in 2002 and 2005, both in computer science. He is currently a professor with the School of Information Science and Engineering of Central South University. He is also a member (E200012141M) of the China Computer Federation (CCF). His major research interests are service computing and wireless sensor networks.

In the future, the ability to obtain data will grow rapidly, and the growth of network transmission capacity will be slow; the contradiction between the two will become increasingly prominent. We believe that a fundamental solution is to construct a network that transmits the information after refining data in the future network.