Received June 2, 2016, accepted June 12, 2016, date of publication June 21, 2016, date of current version July 7, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2582748

Traffic Engineering in Software-Defined Networking: Measurement and Management

ZHAOGANG SHU[1], JIAFU WAN[2], JIAXIANG LIN[1], SHIYONG WANG[2], DI LI[2], SEUNGMIN RHO[3], AND CHANGCAI YANG[1]

[1]College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China
[2]School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou 510641, China
[3]Department of Media Software, Sungkyul University, Anyang, South Korea

Corresponding author: J. Wan (jiafuwan_76@163.com)

This work was supported in part by the National Natural Science Foundation, China, under Grant 61572220, Grant 41401458, Grant 61262013, and Grant 51575194, in part by the Natural Science Foundation of Fujian Province, China, under Grant 2016J05148 and Grant 2014J05045, in part by the Education Department Foundation of Fujian Province, China, under Grant JA14119, in part by the Natural Science Foundation of Guangdong Province, China, under Grant 2016A030313734 and Grant 2016A030313735, in part by the Fundamental Research Funds for the Central Universities under Grant 2015ZZ079, in part by the National Key Technology Research and Development Program of China under Grant 5642015BAF20B01, and in part by the Science and Technology Planning Project of Guangdong Province, China, under Grant 2014B090921003 and Grant 2015B010101005.

ABSTRACT As the next generation network architecture, software-defined networking (SDN) has exciting application prospects. Its core idea is to separate the forwarding layer and control layer of network system, where network operators can program packet forwarding behavior to significantly improve the innovation capability of network applications. Traffic engineering (TE) is an important network application, which studies measurement and management of network traffic, and designs reasonable routing mechanisms to guide network traffic to improve utilization of network resources, and better meet requirements of the network quality of service (QoS). Compared with the traditional networks, the SDN has many advantages to support TE due to its distinguish characteristics, such as isolation of control and forwarding, global centralized control, and programmability of network behavior. This paper focuses on the traffic engineering technology based on the SDN. First, we propose a reference framework for TE in the SDN, which consists of two parts, traffic measurement and traffic management. Traffic measurement is responsible for monitoring and analyzing real-time network traffic, as a prerequisite for traffic management. In the proposed framework, technologies related to traffic measurement include network parameters measurement, a general measurement framework, and traffic analysis and prediction; technologies related to traffic management include traffic load balancing, QoS-guarantee scheduling, energy-saving scheduling, and traffic management for the hybrid IP/SDN. Current existing technologies are discussed in detail, and our insights into future development of TE in the SDN are offered.

INDEX TERMS Software-defined networking, SDN, traffic engineering, network monitoring, network measurement, network management.

I. INTRODUCTION

With the continuous development and in-depth application of cloud computing and Internet of Things (IoT) [1], [2], the traditional network architecture cannot meet the requirements of current industry fields, such as Cyber-Physical Systems(CPS) [3], 5G wireless network [4], and Internet of Vehicle [5]. Therefore, some researchers proposed that Software Defined Network (SDN) will be applied to the industrial 4.0 environment, which will increase flexibility and innovation capacity of IoT in industrial system [6]–[8].

Security and Traffic Engineering (TE) are both research topics for SDN applications in the fields of industrial environments. In the literature [9], we studied security issues of SDN, in this paper, we will mainly focus on the TE problem of SDN. TE is an important application related to network systems, whose main task is to study how to measure and analyze real-time network traffic, and design reasonable routing mechanisms to schedule and guide network traffic to improve utilization of network resources, or better meet requirements of the network Quality of Service (QoS). Traditional network

TE technologies mainly include IP-based TE and MPLS-based TE.

In general, IP-based TE solves the problem of multipath traffic load balance by optimizing the IP routing algorithm to avoid network congestions [11]. For example, Fortz and Thorup [10] propose a neighborhood search algorithm, which is based on link weights of Open Short Path First (OSPF) to adjust the routing calculation strategy, and finally get multiple equivalent shortest paths to achieve traffic load balance. Chen et al. design a multipath planning for IoT multimedia sensing [12]. IP-based TE technology has two clear drawbacks: first, when OSPF link weights are used to control routing of a network, traffic cannot be split in an arbitrary proportion, leading to inability to make full use of network resources; second, when links fail, or link weights of the network topology change, the OSPF protocol takes some time to converge to a new network topology, which possibly leads to network congestions, packet losses, delays, and even routing loops.

In order to avoid these defects of IP-based TE, researchers proposed another solution, in which network packets are forwarded by the Multi-Protocol Label Switching (MPLS) [13], instead of IP headers. However, the protocol mechanism of the MPLS is too complex, and can lead to a high performance overhead, so it is difficult to satisfy requirements of data center networks demanding high link bandwidth utilization, green energy saving, and high reliability.

To sum up, control management and data forwarding in traditional networks are tightly coupled, since the whole network is controlled by distributed devices, where fine-grained control of traffic management cannot be achieved, and flexibility and extensibility are hard to improve. Therefore, it is necessarily to develop a network architecture and corresponding TE technology to solve this problem. Software-Defined Network (SDN) is an innovative network architecture that was proposed by researchers of Stanford University, and received widespread attention in recent years. Its core idea is to separate the forwarding and control planes of a network system [14], [15], so that network operators can program packet forwarding behavior to significantly improve the innovation capability of network applications. Compared to the traditional network architecture, SDN has the following distinguish characteristics.

1) Concentration of control. SDN controller stores the entire network information, including the network topology, dynamic changes of the network status, and global application requirements, such as QoS and security requirements [16].

2) Programmability. A network operator can dynamically program data forwarding layer devices to optimize the allocation of network resources.

3) Openness. Forwarding equipment has a unified interface to communicate with SDN controller, which does not depend on different equipment suppliers, and the SDN controller can conveniently obtain network status data to schedule network traffic.

The characteristics of SDN are useful for solving current problems of network traffic engineering, which can be summarized as follows [17].

1) Traffic measurement. We can deploy scalable global measurement tasks flexibly in the SDN, which can collect real-time network status information, and monitor and analyze traffic centrally in the controller.

2) Traffic scheduling and management. Traffic application requirements can be considered globally, so that flexible, granular traffic scheduling is possible.

3) The OpenFlow switch has multiple stream table pipelines, which make flow management more flexible and efficient.

However, although SDN provides much support for traffic engineering, it has many problems that are needed to be solved. Currently, there is no research that would show that the traditional network TE technique can be fully compatible with the SDN [17]. Another problem is that SDN and traditional IP network will coexist for a long time, and there is little research considering the TE technology of hybrid IP/SDN networks [18]. Therefore, TE technologies based on SDN is of great significance for further applications of SDN.

The rest of paper is organized as follows. Section II proposes a framework for TE in the SDN, describing its components, and relationships among them. Section III discusses traffic measurement issues related to the SDN in detail. Section IV discusses traffic measurement issues related to the SDN and IP/SDN in detail. In Section V we discuss related open issues of TE in the SDN, and provide conclusions.

II. FRAMEWORK FOR TE IN SDN

Combining the ideas of TE for traditional networks with characteristics of the SDN, we propose a framework for TE in the SDN, as illustrated in Figure 1.

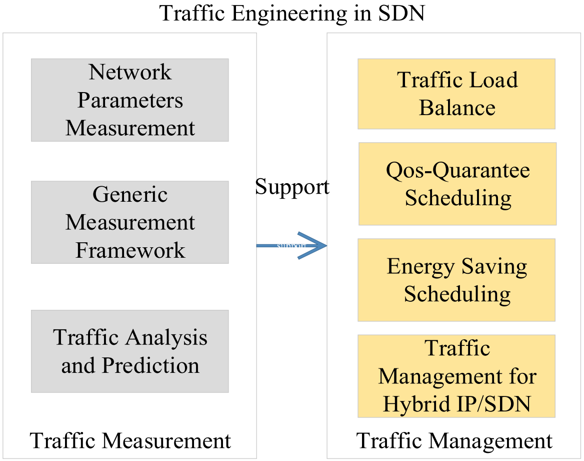
Traffic Engineering in SDN
Network Parameters Measurement
Generic Measurement Framework
Traffic Analysis and Prediction
Traffic Measurement
Support
Traffic Load Balance
Qos-Quarantee Scheduling
Energy Saving Scheduling
Traffic Management for Hybrid IP/SDN
Traffic Management

FIGURE 1. Framework for TE in SDN.

The framework includes two parts: traffic measurement and traffic management. Traffic measurement mainly studies how to monitor, measure and acquire network status information in the SDN environment. The network status information includes the current topology connection status, ports' status (up or down), various kinds of packet counters, dropped packet counters, utilization ratios of link bandwidths, end-to-end network latency, end-to-end traffic matrices and so on. Based on the network status information, we can validate whether the current network status is correct, and predict the future traffic trend by analyzing packet counters statistics, to avoid network congestions and improve network efficiency. We divide research work on traffic measurement into three directions: network parameters measurement, a generic measurement framework, traffic analysis and prediction. Traffic management mainly studies how to manage and schedule network traffic based on the network status information provided by the traffic measurement technology, to satisfy end user requirements of network applications, such as QoS. We divide research work on traffic management into four directions: traffic load balancing, QoS-guarantee scheduling, energy-saving scheduling, and traffic management for the hybrid IP/SDN. It is worth mentioning that the hybrid IP/SDN exists for a long time. Therefore, it is necessary to study TE in the hybrid IP/SDN.

III. TRAFFIC MEASUREMENT IN SDN

Traffic measurement is an important research branch of TE. This section discusses related technologies from the following three aspects: network parameters design and monitoring technologies, a universal framework of flow measurement, traffic analysis and prediction technologies.

A. NETWORK PARAMETERS MEASUREMENT

Network parameters are a set of indices that represent the current network status. Reasonable network parameter design is a precondition for effective network management. Therefore, network parameters design is a primary task of network measurement. We think that SDN network measurement parameters mainly include three types: network topology parameters, network traffic parameters, and network performance parameters.

Network topology parameters refer to the number of network nodes, link bandwidths, connection structures and port statuses. According to the OpenFlow specification, the SDN controller detects the current network topology actively using Link Layer Discovery Protocol (LLDP) [19], and maintains global topology information. During the process of link discovery, the controller sends LLDP packets as Packet-out messages to all switches in the network. When the SDN switch receives an LLDP packet from the controller, it sends the packet to all other switches connected to it directly. When a switch receives an LLDP packet from another switch, it sends the LLDP packet to the controller as a Packet-in message for help, because there is no corresponding forwarding rule in the switch's FlowTable. After the SDN controller receives Packet-in messages with LLDP, it can analyze which switches are connected directly to each other, and construct the global topology [20]. The corresponding working process is illustrated in Figure 2.

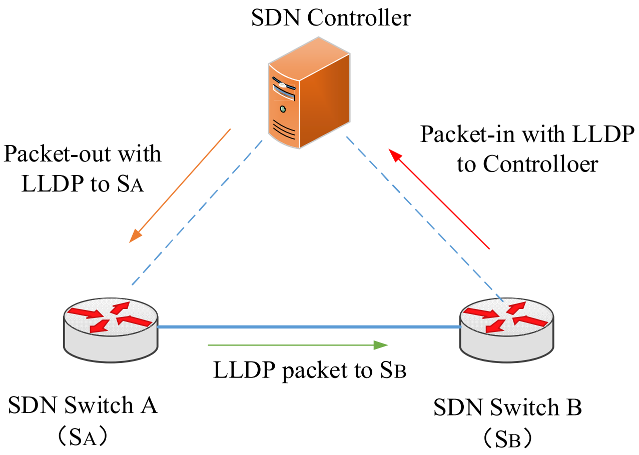

FIGURE 2. Detection process using LLDP for SDN topology.

Network traffic parameters refer to the number or speed of network packets that pass through network equipment or a network port, such as the total number of IP packets, the number of bytes per second and so on. Network traffic parameters are considered as an important basis for detection of the current network status and analysis of network user behavior. In the SDN network, there are two types of network traffic, control traffic and data traffic. Control traffic is a data flow that is transmitted between the SDN controller and switches, and data traffic is a data flow that is transmitted between switches. In order to determine flow characteristics, we must collect statistical information on each switch port, including the number of packets, size of packets, as well as the end-to-end traffic matrix of the entire network, which represents the network flow between any two network nodes.

Yuan et al. [21] propose a programmable measurement architecture, namely, the ProgME, which can collect user-specific traffic statistics. Tootoonchian et al. [22] propose a traffic matrix estimation system for the SDN, which is called the OpenTM. The OpenTM can detect all active network flows according to the routing information of the controller and flow forwarding path information. The OpenTM includes various selective querying methods for routing nodes to obtain accurate information about the packets number and flow evaluation.

In general, the number of network flows is very big, but available measurement resources, such as the CPU and Ternary Content Addressable Memory (TCAM) table, are limited, so it is impossible to obtain the traffic matrix by measuring the size of each flow directly. In order to solve this problem, the iSTAMP method [23] first measures the size of the largest k flows or the convergence flow, and then estimates the integrated traffic matrix using relevant technologies. The iSTAMP seems to find balance between limitations of network resources and measurement accuracy, but it still has the following drawbacks. First, when SDN switches forward packets, the matching principle to find a forwarding item is based on priority and wildcards, and only flows having same destination node can be aggregated into a single item, implying feasibility constraints of flow aggregation. But the iSTAMP method ignores these constraints in aggregation of flows. Second, in order to find the largest k flows, the iSTAMP uses the TCAM table to measure single flows in multiple time periods, which increases measurement costs that cannot be ignored. Hu and Luo [24] study traffic matrix estimation in a data center network based on the SDN. It assumes that the aggregation of flows occurs only in ToR SDN switches, to avoid the flow aggregation problem mentioned above. However, it is difficult to implement the method in a real network.

Network performance parameters mainly include the network latency, bandwidth utilization, throughput rates, packet loss rates etc. Van Adrichem et al. [25] propose a network monitoring module, the OpenNetMon, which designs several measurement methods for network performance parameters. The OpenNetMon can monitor network throughput, packet loss rates, and network delays continuously. First, when it measures the throughput rate, it only queries the last switch on the forwarding path using an adaptive frequency. The counter returns the number of packets (S) of each flow in the sampling interval (T), and the forwarding path throughput can be obtained as S/T. Second, when it calculates packet loss rates, it queries the first and last switches on the forwarding path to obtain counter statistics, then the increment of the first counter minus the increment of the last counter in a measurement cycle, and finally the result is divided by the time, giving the packet loss rate of this forwarding path. Third, to calculate the delay, the OpenNetMon sends probe messages to data layers of the forwarding path, these messages get through all nodes of the path, and finally return to the controller, where the communication delays can be obtained by calculating time differences. The FlowSense [26] develops a monitoring module for the controller of the SDN, which can analyze dynamic changes in network flows according to messages received by the controller. For example, after the controller receives a FlowRemoved message for a certain flow, it obtains the throughput rate of this flow using the statistical values of the flow divided by the size of the corresponding flow.

B. GENERIC MEASUREMENT FRAMEWORK

Network traffic measurement has two typical modes, active and passive modes. The active measurement mode produces extra detecting traffic, while the passive measurement mode does not, because it only monitors traffic that passes through SDN switches. Many SDN measurement systems adopt traffic monitoring methods of traditional IP networks based on packet sampling, in which SDN switches randomly crawl local packets to obtain traffic statistics. A famous system for packet sampling and analysis is the NetFlow developed by Cisco [27]. The NetFlow identifies flows using five-tuples, where a switch maintains cache record information for each flow. When a flow arrives, the NetFlow compares the head of this flow with the information stored in a local cache record. If they match, it updates the package count; otherwise it adds a new entry to the cache record for this new flow. Other flow sampling solutions include the sFlow [28] and JFlow [29]. They do sampling at the entrance of each router in a certain proportion, and then send head information of the flow and corresponding timestamp to the center collector server. All above measurement methods imply a lot of burden on the SDN central controller in large-scale networks. Therefore, they are not suitable for SDN applications.

In order to support a variety of measurement tasks and applications for the SDN effectively, it is necessary to design a general SDN measurement framework. The PayLess [30] is a flow measurement framework with low costs based on polling. The PayLess framework is designed as a component of the OpenFlow controller, and provides a RESTful style interface for applications. In particular, the PayLess is responsible for parsing request commands from the application level of tasks, and transforming these commands into path planning on some switches. Therefore, the PayLess maintains an abstract view of network information for applications, and provides unified programming interfaces for a variety of network applications. An application defines a task in JSON format, where custom components of users can be added to the measurement framework. Yu et al. [31] present a general, abstract measurement framework, the OpenSketch, using a software-defined method to measure traffic. Flow-based measurement has defects such as counter overhead and measurement inaccuracy, so it is unable to meet requirements of different users. On the other hand, the OpenSketch supports customization for different users, and different tasks have different requirements for hardware, so it is hard to deploy multiple tasks on the same hardware equipment. In order to fully consider versatility and efficiency, the OpenSketch proposes that measurement control and data layers should be separated, where data layers are designed as three-phase actions that can be configured dynamically. At the first phase,the OpenSketch offers a variety of hash algorithms that map packets into a small amount of measurement data. At the second phase, it stores and search rules using the TCAM and wildcard matching. At the third phase, it provides flexible counting using the SRAM. Using this pipeline design, the OpenSketch abstracts a variety of measure algorithms into several general steps. At the same time, it provides various measurement modules in the control layer which allows users to adjust any phases of the data layer mentioned above. But it must be noted that the OpenSketch needs to redesign some parts of hardware, which possibly restricts its widespread application. Figure 3 shows the architecture of the OpenSketch.

Traffic information collection and aggregation are two important parts of traffic measurement, which need hardware support in the data layer, such as the TCAM. However, the existing hardware resources of SDN switches are often limited. In addition, network applications have multiple simultaneous flow monitoring task requirements, instead of

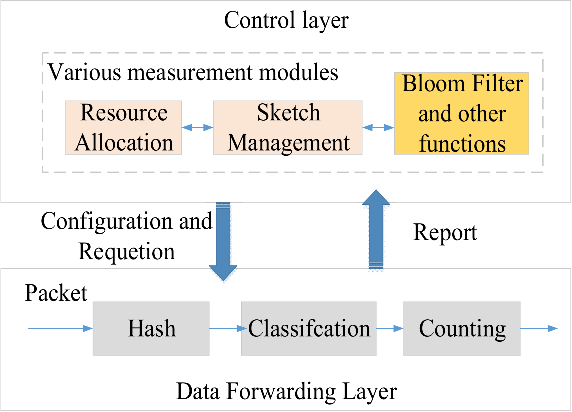

FIGURE 3. Architecture of OpenSketch.

supporting than only one measurement task at a period of time. Therefore, the question of how to allocate hardware resources reasonably to provide accurate and effective flow measurement is also very important. From this point of view, Moshref et al. [32] propose an adaptive management architecture for hardware resources, the DREAM, which can effectively balance measurement accuracy and resource costs. The core ideas of the architecture are as follows.

1) The accuracy of a flow measurement task depends on the hardware resources assigned to the task. With an increase in the number of resources assigned to the task, there will be a point, after which the ability of promoting task accuracy becomes smaller with further increase in resources.

2) Resources needed by measurement tasks are different depending on the time and size of the flow, so it is possible to adjust these resources to raise their utilization efficiency. The DREAM includes three levels: the top level is the user layer that is responsible for generating measurement tasks, including types of task, specific flow thresholds, accuracy requirements etc. The middle level is the DREAM algorithms running on the SDN controller. They receive tasks from users, then create corresponding task objects, and finally deploy these abstract tasks into multiple switches. In fact, the deployment process is also the process to request TCAM resources in the switch, and to store measurement logic in the TCAM. According to the real-time feedback of traffic monitoring, the DREAM algorithms can evaluate the accuracy of measurements, and adjust resources allocated to each task, or reuse resources of multiple tasks. On the other hand, the algorithms can decide whether to accept a new task according to the current situation. The bottom level is SDN forwarding devices that are responsible for measuring storage resources, and return results of real-time flow counting.

C. TRAFFIC ANALYSIS AND PREDICTION

Traffic analysis and prediction is an important part of network measurement, and one of their goals is to distinguish network anomalous traffic, and to analyze possible contingencies in the network, such network congestions. This way, we can provide better data for network traffic scheduling and management. On the other hand, another important object is to test whether the current status of the network is correct, such as correctness of current network equipment configurations, routing loop problems etc. In this section, we introduce related technologies with respect to SDN traffic analysis and prediction.

Mehdi et al. [33] reconsider the application of several traditional abnormal traffic detection methods to the SDN, and provides a good practical deployment experience in SDN anomalous traffic detection. However, these methods are relatively simple, and only suitable for some specific scenarios. Zuo et al. [34] propose an online traffic anomaly detection method, the OpenTAD. First, the flow table statistic is collected from the controller online, and the traffic matrix and sample entropy matrix are constructed and assembled. Then the PCA method is used to detect abnormal traffic. The OpenTAD is simple and effective, and traffic anomalies can be isolated rapidly. This method is a lightweight online traffic anomaly detection method for the SDN.

Data forwarding in networks depend on all kinds of network equipment providing a variety of functions, and the process is complex and vulnerable to potential configuration errors, software bugs, and other issues, which possibly cause various network mistakes, such as routing loops. The SDN simplifies network applications, but it is still difficult to avoid mistakes due to complexity of the software itself. Therefore, correct status detection and validation is an important research task in the field of traffic measurement for the SDN.

Mai et al. [35] propose a correct status detection tool for the SDN based on static analysis of data layers, called the Anteater. The main working processes of the Anteater are as follows. First, it converts network connection states and switches' Forward Information Base (FIB) to Boolean expressions, and then transforms three kinds of correct statuses, namely, loopless routing, network connectivity, and consistency of rules, into the expression of the SAT problem. Second, choosing a correct status waiting to be checked, it combines the status and above Boolean expressions and input into the SAT solver for analysis. If there is a configuration rule that damages the correct status, the Anteater constructs a special packet to trigger the corresponding problem. In actual tests, the Anteater successfully deployed in a campus network found 23 network problems. In this paper, the author points out that diagnosis should be close to the network data level as much as possible, so as to observe network behavior more directly, and find problems before the configuration takes effect. This way, the Anteater also supports unified analysis for multiple protocols and language environments.

Canini et al. [36] propose a network validation tool combining model checking with symbolic execution, the NICE, which is developed in the Python programming language to seamlessly support the OpenFlow control program. Its main purpose is to find errors in the OpenFlow application. For example, the controller responses to a new flow and produces a new rule. At this point, if excess flows also arrive at the controller, application errors may happen. The NICE considers the OpenFlow program and network correctness status as input parameters, and verifies the correctness of each state by state space search using model check, and outputs the final correctness state report. With the NICE, the OpenFlow program can inspect routing loops and other network problems.

Khurshid et al. [37] proposes SDN-oriented real-time checking and debugging tools, the VeriFlow. The tools are inserted between the SDN control layer and data layer, and they can monitor and inspect each configuration rule. Configuration conflicts can be detected in advance, and warnings can be sent timely. When the VeriFlow updates or deletes rules, the VeriFlow validates the rules using three steps. First, packets with the same rules and actions are organized in an equivalence class that is validated together as an atomic unit. Second, a forwarding figure is created for the equivalence class. Finally, the updated rules and corresponding forwarding figure are inputted, and all equivalence class paths are depth-first traversed to determine whether a rule update will produce network anomalies. Experiment results show that the VeriFlow can finish network status inspection within hundreds of microseconds.

IV. TRAFFIC MANAGEMENT IN SDN

Network management aims to maintain network availability and improve network performance. Reasonable scheduling of network traffic is an important way to improve QoS of the network. In general, there are multiple paths between the source and destination node in the SDN, which provide possibility for traffic scheduling. The SDN controller maintains the global view of present use of each path in the network using various network measurement technologies mentioned above, and we can design a traffic scheduling algorithm to dynamically plan data forwarding paths to meet users' requirements. In this section, we discuss traffic management technologies in the SDN from the following aspects: traffic load balancing, QoS-guarantee scheduling, energy-saving scheduling, and traffic management for the hybrid IP/SDN.

A. TRAFFIC LOAD BALANCE

In the SDN, there are two types of network traffic, data layer traffic and control layer traffic. In this section, we mainly discuss the load balancing technology for data layer traffic. Compared to the traditional network, the main advantage of load balancing in the SDN is that forwarding decision calculations are centralized and not distributed, allowing considering multiple optional link utilization rates and flow characteristics more comprehensively, to plan the strategy of load balancing better.

The equivalence multipath routing technology (ECMP) based on a hash algorithm is an effective load balancing solution. According to ECMP routing, there may be multiple forwarding paths for a target network. When the packet arrives at a switch or router, switch or router extracts the header fields of the packet to make hash calculations, and then selects one of the forward paths according to the hash value, which results in that IP packets having the same head are forwarded along the same path. A clear defect of the ECMP is that a lot of large flows, called elephant flows, are forwarded to the same path, which results in load imbalance and bandwidth waste [38], as shown in Figure 4.

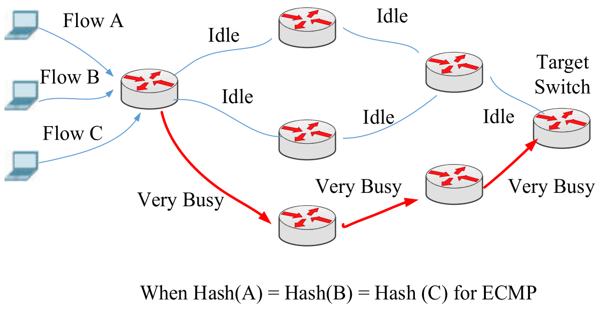

FIGURE 4. Senario of load imbalance for ECMP.

A simple solution is to split traffic from the package level, not the flow level. This method greatly improves effectiveness of load balancing, but it makes reordering problem of the TCP send window more serious, which causes unnecessary TCP send window shrinkage [39]. In order to solve the above problem, several improved ECMP schemes are proposed [40]–[42]. Their main idea is to identify elephant flows first, and then choose the right path by the controller. Typical solutions include Hedera [43], Mahout [44], and MicroTE [45].

Hedera [43] is a dynamic and scalable traffic management system, which schedules packet flows adaptively to effectively utilize network resources, and tries to realize equal bandwidths. Hedera also points out that in order to effectively utilize multiple paths between data center severs, we must detect and manage elephant flows. Based on this viewpoint, the scheduling strategy of Hedera contains three basic steps. First, it detects elephant flows in the edge switch and switch forward flows using the ECMP method by default. Second, Hedera collects flow information of switches by periodically inquiring them, and if a flow rate is larger than a specific threshold, then it is marked as an elephant flow. Third, Hedera estimates bandwidth requirements of those elephant flows. According to these requirements, combined with the current load situation, Hedera calculates a reasonable path dynamically for elephant flows.

Mahout [44] is another system to manage and optimize traffic in the SDN, which reduces the cost of traffic management using an additional backend server. Because detection of elephant flows in Hedera is based on internal network detection, which tends to bring long latency and higher resource overhead, Mahout uses a backend server rather than forwarding equipment to detect elephant flows, which can improve management efficiency and reduce management costs. Mahout abstracts a special function layer of the operating system of the backend server, namely, the SHIM layer, which is responsible for monitoring local traffic by a socket buffer. When the buffer exceeds a specified threshold, Mahout thinks that the corresponding flow is an elephant flow, and subsequent packets are marked. Mahout defines priorities for rules of the flow table, high priority and low priority. By default, packets matching the rules of low priority are forwarded using the ECMP. The packets of an elephant flow that match the rules of high priority are sent to Mahout to calculate optimal routing, and the corresponding forwarding rules in the switch are updated.

MicroTE [45] is a solution similar to Mahout. MicroTE monitors the flow status by flow statistics passively. When the flow status is clearly changed, it triggers flow aggregation behavior. Depending on the difference of the size of the average flow rate and instantaneous flow rate, MicroTE judges whether the current flow is an elephant flow or it can be predicted. When the flow rate can be predicted, MicroTE starts calculation of routing optimization, or deals with it using a heuristic ECMP algorithm.

Another traffic load balancing mechanism is based on the wildcard scheduling technology, where two typical solutions are the DevoFlow [46] and DIFANE [47]. The DevoFlow modifies the OpenFlow model to develop a resource-saving and scalable traffic scheduling scheme. Specifically, the DevoFlow proposes that the controller should reroute elephant flows only after finding them, and not regard all data flows as potential elephant flows, which can effectively avoid delays and controlling costs. The DevoFlow designs a multipath matching rule based on wildcards. By default, a switch uses this rule to forward flows. At the same time, The DevoFlow introduces a method of traffic statistics to identify elephant flows, where elephant flows need help of the controller to be rerouted. The DIFANE [47] is another efficient traffic scheduling scheme based on wildcards. The core idea of the DIFANE can be summarized in two aspects. First, the controller distributes some rules to an authorized subset of switches. Second, switches handle all packages in the data layer. If flows cannot match local rules, they are encapsulated and redirected to an appropriate authorized subset of switches.

B. QoS-GUARANTEE SCHEDULING

In a network, a lot of real-time business traffic, such as voice data, instant messages and so on, is sensitive to delays and packet losses in the process of transmission. Therefore, reasonable scheduling of network resources to provide QoS for business is an important problem of traffic management. The SDN provides an open control interface to support flexible network traffic scheduling strategies, which can satisfy QoS requirements of different network applications.

Yan et al. [48] propose a QoS-guarantee solution in the SDN, the HiQoS. The HiQoS identifies multiple paths between source and destination nodes by queuing mechanisms to guarantee QoS for different types of traffic. Experimental results show that the HiQoS scheme can reduce delays and increase throughput. It is worth mentioning that the HiQoS can recover from link failure very quickly by rerouting traffic from failed paths to other available paths.

The OpenQoS scheme [49] aims to provide QoS guarantee for multimedia business flows distribution. Because packet heads of multimedia business flows are very different to other packet heads, the OpenQoS divides all data traffic into two groups, multimedia and data flows, using OpenFlow configuration matching rules. The OpenQoS observes performance of forwarding paths in terms of delays and packet losses, and selects a best path that can satisfy QoS requirements. Other data flows are still forwarded to the original path. But the OpenQoS only optimizes multimedia flows scheduling, and does not consider business flows having multiple QoS requirements.

Ongaro et al. [50] present QoS an enhancement framework based on centralized traffic scheduling for the SDN. The specific questions that the framework is focused on include the following. First, in a network having overlapping nodes or links, how to select appropriate paths for business flows of multiple applications. Second, under requirements of QoS, the sum of bandwidths that are assigned to each path cannot exceed physical link capacity. In order to solve these problems, the framework designs several key components, namely, a network topology mapping module, network status collection module, path selection module, and dynamic path configuration module. The first two modules are responsible for monitoring the data layer to capture topology updates, collect network parameters dynamically, and generate weights of each path in the network diagram. Path selection module chooses appropriate paths according to QoS requirements of business and weights of the network diagram. Dynamic path configuration module is responsible for updating routing rules in the data forwarding layer at the right time. As the core module of the framework, the path selection module makes scheduling decisions using the model of the Multi-Commodity Flow and Constrained Shortest Path (MCFCSP). The model establishes a forwarding strategy in the network G = (N, A) for the business flow set K, and the objective function of the model is the length of link delays and size of packet losses. Based on constraints of parameter variables provided by the model, we can solve the NP complete problems to obtain a global scheduling strategy. Authors verified that this solution based on the Mininet platform can satisfy QoS requirements of real-time business applications, including video and file transfers, effectively.

C. ENERGY-SAVING SCHEDULING

Due to economy and environmental protection, reducing energy consumption has become an important issue in recent years. Information and communication technologies are widely used in everyday life. Energy consumption due to telecommunication networks accounts for 5% of all energy consumption in developed countries, and it is growing at the annual rate of 10% [51], [52]. Therefore, the question of how to reduce network energy consumption is very meaningful. In recent years, many solutions regarding network energy-saving were proposed [53]. Because servers are currently main consumers of energy, there is less research to solve the problem of energy consumption of data center networks [54]. However, the percentage of network energy consumption may even increase to 50%. Therefore, the SDN should not only consider link utilization, load balancing, and QoS requirements, but also energy consumption requirements in the process of traffic management. As is well known, network energy consumption mainly depends on network units, such as links, switches etc. Currently, there are two types of methods to save energy, link rate adaptive and sleep models. Link rate adaptive methods dynamically adjust link rates according to traffic demand, because the energy consumption of links depends on their data transmission loads, rather than utilization [55], On the other hand, the sleep model powers off some network components, or translates some non-working components into sleep mode to save energy.

Ke et al. [56] propose an energy-saving method based on priority scheduling for the SDN, in which the Iterative Parallel Clustering Algorithm (IPGA) is deigned. Compared to the scheduler without energy-saving, the IPGA scheduling can save nearly 30% of energy. The question of the priority scheduling problem is to find those flows that have the highest QoS attributes and use minimum switches. The flows with high priorities are transmitted preferentially, because they need fewer switches, so that we can shut down some additional switches to achieve the goal of reducing power consumption.

Li et al. [57] propose an energy-saving strategy based on exclusive routing. The strategy does not need to synchronize time of servers, and it can manage global flow states using the SDN, and explicitly inform virtual machines to permit or deny a specific flow. When a new flow arrives at a virtual machine of the server, then virtual machine reports to the SDN controller. The SDN controller performs an exclusive routing algorithm, and updates the activation and suspension flows. If a new flow is activated, the SDN controller informs the virtual machine to permit this flow, and writes corresponding items into the FlowTable of the switch. If the activation flow changes into the suspended state, then the SDN controller informs the virtual machine to deny this flow. The principle of exclusive routing is shown in Figure 5.

According to the exclusive routing algorithm, each activation flow monopolizes a routing link, so it does not necessary reserve bandwidth for a particular flow. Network energy saving mainly depends on high utilization of rates of links. For the exclusive routing algorithm, if there are idle paths, flows choose these paths. Otherwise, if a flow has a lower priority, then the flow is suspended.

Amokrane et al. [58] present an online routing selection method, which aims to minimize energy consumption of an access points wireless mesh network (called mesh access points, MAPs). This method can be easily combined with the SDN, because it relies on a central controller, which monitors and manages the entire network, and decides routing paths. This paper considers this problem as an integer linear planning problem. As the objective function, it considers the cost of sleep and active states of MAPs, and also takes into account rerouting or merging of existing flows. Since this problem is an NP-hard problem, the author presents an ant colony algorithm (ACOFER) that is simple but effective. Compared to other algorithms, the ACOFER leads to better effectiveness in terms of energy saving.

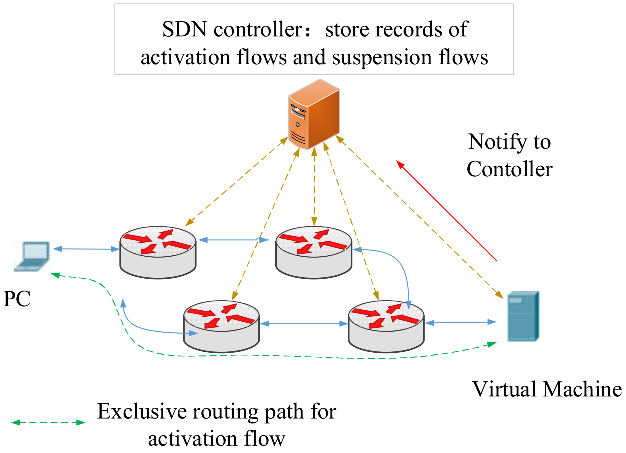
SDN controller：store records of activation flows and suspension flows
Notify to Contoller
PC
Virtual Machine
Exclusive routing path for activation flow

FIGURE 5. Exclusive routing framework for SDN.

D. TRAFFIC MANAGEMENT FOR HYBRID IP/SDN

SDN-based TE technologies provide effective solutions for network optimization. However, the process to replace traditional networks for the SDN lasts a long time, so IP/SDN hybrid network emerged. Not every node of an IP/SDN hybrid network has features of an SDN switch, and we only obtain an accurate global view of the network partially. Compared to the pure SDN, traffic management of the IP/SDN hybrid network is more difficult. Therefore, the question of how to manage traffic in the IP/SDN hybrid network becomes a challenge.

Agarwal et al. [18] present an architecture of traffic manage that is suitable for the IP/SDN hybrid network. In particular, it tries to leverage the centralized controller to obtain high network utilization, as well as to reduce packet losses and delays. First, SDN controller's optimization problems are formulated for traffic engineering with partial deployment. Then the Fully Polynomial Time Approximation Schemes (FPTAS) is designed to solve these problems. Using these algorithms, experiments show that performance gains are achievable in the IP/SDN hybrid network environment.

Guo et al. [59] explore traffic engineering issues in SDN/OSPF hybrid networks. It is assumed that OSPF weights and flow splitting ratios of the SDN switch can both be changed, and the controller can split flows coming into the SDN switch arbitrarily. Traditional network nodes still run the OSPF. A novel algorithm, the SOTE, is proposed to obtain lower maximum link utilization. Results show that when only 30% of SDN nodes are deployed, near optimal performance is obtained.

Guo et al. [60] search for an optimal migration sequence of legacy routers to SDN-enabled routers, so that we can decide where and how many routers to migrate. The authors propose a heuristic algorithm to find a migration sequence of routers that obtains most of benefits from the perspective of traffic engineering. The algorithm was evaluated by simulations. The experiments show that the proposed algorithm is faster than other migration algorithms in searching for a migration sequence. If network nodes can be properly deployed, almost 40% of routers lead to most of benefits.

V. CONCLUSIONS

The development of the traffic engineering technology of the SDN is an important aspect to promote widespread application of the SDN. A successful example is Google B4 network. B4 network takes advantages of characteristics of the SDN to implement the objectives of TE requirements that traditional networks cannot realize. In this paper, we propose a framework for TE in SD and discuss related technologies from two aspects: traffic measurement and traffic management. Regarding network measurement, there are related network parameters in each layer of the network that represent the status of network. In general, we monitor the parameters of the application layer and network layer. The traffic matrix is ideal information to reflect the current state of the network, but predicting technologies of the traffic matrix need further research. Regarding network management, the main goal is to provide reasonable traffic scheduling strategies according to network parameters obtained using flow measurement technologies to satisfy specific requirements. We discuss four aspects, namely, traffic load balancing, QoS-guarantee, energy-saving scheduling, and traffic management of the IP/SDN hybrid network. In fact, there are some issues that are not involved, such as TE based on the MPLS in the SDN, and performance analysis of TE, which are likely to become hot topics in the future.

REFERENCES

[1] J. Wan, C. Zou, K. Zhou, R. Lu, and D. Li, "IoT sensing framework with inter-cloud computing capability in vehicular networking," *Electron. Commerce Res.*, vol. 14, no. 3, pp. 389–416, Nov. 2014.

[2] J. Wan, D. Zhang, Y. Sun, K. Lin, C. Zou, and H. Cai, "VCMIA: A novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 153–160, 2014.

[3] J. Wan, H. Yan, Q. Liu, K. Zhou, R. Lu, and D. Li, "Enabling cyber-physical systems with machine-to-machine technologies," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 13, nos. 3–4, pp. 187–196, 2013.

[4] K. Lin, W. Wang, X. Wang, W. Ji, and J. Wan, "QoE-driven spectrum assignment for 5g wireless networks using SDR," *IEEE Wireless Commun.*, vol. 22, no. 6, pp. 48–55, Dec. 2015.

[5] J. Wan, H. Yan, D. Li, K. Zhou, and L. Zeng, "Cyber-physical systems for optimal energy management scheme of autonomous electric vehicle," *Comput. J.*, vol. 56, no. 8, pp. 947–956, 2013.

[6] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, "Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination," *Comput. Networks.*, vol. 101, pp. 158–168, Jun. 2016.

[7] S. Wang, J. Wan, D. Li, and C. Zhang, "Implementing smart factory of industrie 4.0: An outlook," *Int. J. Distrib. Sensor Netw.*, vol. 4, Jan. 2016, Art. no. 3159805.

[8] J. Wan *et al.*, "Software-defined industrial internet of things in the context of industry 4.0," *IEEE Sensors J.*, doi: 10.1109/JSEN.2016.2565621.

[9] Z. Shu, J. Wan, D. Li, J. Lin, A. V. Vasilakos, and M. Imran, "Security in software-defined networking: Threats and countermeasures," *Mobile Netw. Appl.*, doi: 10.1007/s11036-016-0676-x.

[10] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc. INFOCOM*, vol. 2. Tel Aviv, Israel, Mar. 2000, pp. 519–528.

[11] G. Han, J. Jiang, N. Bao, L. Wan, and M. Guizani, "Routing protocols for underwater wireless sensor networks," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 72–78, Nov. 2015.

[12] M. Chen *et al.*, "Multipath planning based transmissions for IoT multimedia sensing," in *Proc. IEEE IWCMC*, Paphos, Cyprus, Sep. 2016, pp. 1–10.

[13] G. Swallow, "MPLS advantages for traffic engineering," *IEEE Commun. Mag.*, vol. 37, no. 12, pp. 54–57, Dec. 2000.

[14] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[15] J. Liu, Y. Li, M. Chen, W. Dong, and D. Jin, "Software-defined Internet of Things for smart urban sensing," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 55–63, Sep. 2015.

[16] M. Chen, Y. Qian, S. Mao, W. Tang, and X. Yang, "Software-defined mobile networks security," *Mobile Netw. Appl.*, doi: 10.1007/s11036-015-0665-5.

[17] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in SDN-OpenFlow networks," *Comput. Netw.*, vol. 71, pp. 1–30, Oct. 2014.

[18] S. Agarwal, M. Kodialam, and T. V. Lakshman, "Traffic engineering in software defined networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2211–2219.

[19] W.-Y. Huang, T.-Y. Chou, J.-W. Hu, and T.-L. Liu, "Automatical end to end topology discovery and flow viewer on SDN," in *Proc. 28th Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, May 2014, pp. 910–915.

[20] F. Pakzad, M. Portmann, W. L. Tan, and J. Indulska, "Efficient topology discovery in software defined networks," in *Proc. 8th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Dec. 2014, pp. 1–8.

[21] L. Yuan, C. N. Chuah, and P. Mohapatra, "ProgME: Towards programmable network measurement," *IEEE/ACM Trans. Netw.*, vol. 19, no. 1, pp. 115–128, Feb. 2011.

[22] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: Traffic matrix estimator for OpenFlow networks," in *Passive and Active Measurement*. Berlin, Germany: Springer, Apr. 2010, pp. 201–210.

[23] M. Malboubi, L. Wang, C.-N. Chuah, and P. Sharma, "Intelligent SDN based traffic (de) aggregation and measurement paradigm (iSTAMP)," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr./May 2014, pp. 934–942.

[24] Z. Hu and J. Luo, "Cracking network monitoring in DCNs with SDN," in *Proc. IEEE Comput. Commun.*, Apr./May 2015, pp. 199–207.

[25] N. L. M. van Adrichem, C. Doerr, and F. A. Kuipers, "OpenNetMon: Network monitoring in OpenFlow software-defined networks," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, May 2014, pp. 1–8.

[26] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha, "FlowSense: Monitoring network utilization with zero measurement cost," in *Proc. Int. Conf. Passive Active Meas.*, vol. 1. 2013, pp. 31–41.

[27] (2004). *NetFlow*. [Online]. Available: http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html

[28] (2004). *sFlow*. [Online]. Available: http://www.sflow.org/sFlowOverview.pdf

[29] A. C. Myers, "JFlow: Practical mostly-static information flow control," in *Proc. 26th ACM SIGPLAN-SIGACT Symp. Principles Program. Lang.*, 1999, pp. 228–241.

[30] S. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, "PayLess: A low cost network monitoring framework for software defined networks," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–9.

[31] M. Yu, L. Jose, and R. Miao, "Software defined traffic measurement with OpenSketch," in *Proc. 10th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2013, pp. 29–42.

[32] M. Moshref, M. Yu, R. Govindan, and A. Vahdat, "DREAM: Dynamic resource allocation for software-defined measurement," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 419–430, 2015.

[33] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in *Proc. Int. Symp. Recent Adv. Intrusion Detection*, Menlo Park, CA, USA, Sep. 2011, pp. 161–180.

[34] Q. Zuo, M. Chen, X. Wang, and B. Liu, "Online traffic anomaly detection method for SDN," (in Chinese), *J. Xidian Univ.*, vol. 42, no. 1, pp. 155–160, 2015.

[35] H. Mai, A. Khurshid, R. Agarwal, M. Caesar, P. B. Godfrey, and S. T. King, "Debugging the data plane with anteater," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 290–301, Aug. 2011.

[36] M. Canini, D. Venzano, P. Perešíni, D. Kostić, and J. Rexford, "A NICE way to test OpenFlow applications," in *Proc. 9th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2012, pp. 127–140.

[37] A. Khurshid, X. Zou, W. Zhou, M. Caesar, and P. Godfrey, "VeriFlow: Verifying network-wide invariants in real time," in *Proc. 10th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2013, pp. 15–27.

[38] A. Dixit, P. Prakash, Y. Hu, and R. Kompella, "On the impact of packet spraying in data center networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2130–2138.

[39] M. Chiesa, G. Kindler, and M. Schapira, "Traffic engineering with equal-cost-multipath: An algorithmic perspective," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 1590–1598.

[40] G. Han, Y. Dong, H. Guo, L. Shu, and D. Wu, "Cross-layer optimized routing in wireless sensor networks with duty cycle and energy harvesting," *Wireless Commun. Mobile Comput.*, vol. 15, no. 16, pp. 1957–1981, Nov. 2015.

[41] M. Chen, V. C. M. Leung, S. Mao, and Y. Yuan, "Directional geographical routing for real-time video communications in wireless sensor networks," *Comput. Commun.*, vol. 30, no. 17, pp. 3368–3383, Nov. 2007.

[42] M. Chen, V. C. M. Leung, S. Mao, and M. Li, "Cross-layer and path priority scheduling based real-time video communications over wireless sensor networks," in *Proc. IEEE 67th Veh. Technol. Conf. (VTC)*, Singapore, May 2008, pp. 2873–2877.

[43] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. Netw. Syst. Design Implement. Symp. (NSDI)*, vol. 10. 2010, p. 19.

[44] A. R. Curtis, W. Kim, and P. Yalagandula, "Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection," in *Proc. IEEE INFOCOMM*, Apr. 2011, pp. 1629–1637.

[45] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: Fine grained traffic engineering for data centers," in *Proc. 7th Conf. Emerg. Netw. Experim. Technol. (CoNEXT)*, Dec. 2011, pp. 1–12.

[46] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: Scaling flow management for high-performance networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 254–265, 2011.

[47] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with DIFANE," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 351–362, Oct. 2010.

[48] J. Yan, H. Zhang, Q. Shuai, B. Liu, and X. Guo, "HiQoS: An SDN-based multipath QoS solution," *China Commun.*, vol. 12, no. 5, pp. 123–133, May 2015.

[49] H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end quality of service over software-defined networks," in *Proc. Signal Inf. Process. Assoc. Summit Conf.*, Dec. 2012, pp. 1–8.

[50] F. Ongaro, E. Cerqueira, L. Foschini, A. Corradi, and M. Gerla, "Enhancing the quality level support for real-time multimedia applications in software-defined networks," in *Proc. Int. Conf. Comput., Netw. Commun.*, Feb. 2015, pp. 505–509.

[51] T. Silva and A. Arsenio, "A survey on energy efficiency for the future Internet," *Int. J. Comput. Commun. Eng.*, vol. 2, no. 5, pp. 595–689, 2013.

[52] M. Chen, Y. Zhang, Y. Li, S. Mao, and V. C. M. Leung, "EMC: Emotion-aware mobile cloud computing in 5G," *IEEE Netw.*, vol. 29, no. 2, pp. 32–38, Mar./Apr. 2015.

[53] G. Han, L. Liu, J. Jiang, L. Shu, and G. Hancke, "Analysis of energy-efficient connected target coverage algorithms for industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2015.2513767.

[54] M. Chen, H. Jin, Y. Wen, and V. C. M. Leung, "Enabling technologies for future data center networking: A primer," *IEEE Netw.*, vol. 27, no. 4, pp. 8–15, Jul. 2013.

[55] M. Chen, Y. Hao, Y. Li, C. Lai, and D. Wu, "On the computation offloading at ad hoc cloudlet: Architecture and service models," *IEEE Commun.*, vol. 53, no. 6, pp. 18–24, Jun. 2015.

[56] B.-Y. Ke, P.-L. Tien, and Y.-L. Hsiao, "Parallel prioritized flow scheduling for software defined data center network," in *Proc. IEEE 14th Int. Conf. High Perform. Switching Routing (HPSR)*, Jul. 2013, pp. 217–218.

[57] D. Li, Y. Shang, and C. Chen, "Software defined green data center network with exclusive routing," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 1743–1751.

[58] A. Amokrane, R. Langar, R. Boutabayz, and G. Pujolle, "Online flow-based energy efficient management in wireless mesh networks," in *Proc. Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 329–335.

[59] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, "Traffic engineering in SDN/OSPF hybrid network," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols (ICNP)*, Oct. 2014, pp. 563–568.

[60] Y. Guo, Z. Wang, X. Yin, X. Shi, J. Wu, and H. Zhang, "Incremental deployment for traffic engineering in hybrid SDN network," in *Proc. IEEE 34th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2015, pp. 1–8.

ZHAOGANG SHU received the Ph.D. degree in control theory and control engineering from the South China University of Technology, Guangzhou, China, in 2008. From 2008 to 2012, he was a Senior Engineer with Ruijie Network Communication Corporation, Fuzhou, China. Since 2012, he has been a Lecturer with the College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou. He has directed two research projects and authored over 15 scientific papers and holds four patents. His research interests include software-defined network, network security, and cloud computing.

JIAFU WAN (M'11) is currently a Professor with the School of Mechanical and Automotive Engineering, South China University of Technology, China. His research interests include Industry 4.0, industrial wireless networks, cyber-physical systems, Internet of Things, cloud computing, embedded systems, and industrial robotics. He has directed 12 research projects, including the National Natural Science Foundation of China, the High-Level Talent Project of Guangdong Province, and the Natural Science Foundation of Guangdong Province. Thus far, he has authored/co-authored one book and over 100 scientific papers (with over 50 indexed by ISI SCIE and 40 indexed by EI Compendex) cited over 1700 times. His research results were published in several famous journals, such as the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, the *IEEE Network*, the *IEEE Communications Magazine*, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE WIRELESS COMMUNICATIONS, and the *ACM Transactions on Embedded Computing Systems*. He is a Guest Editor of the IEEE SYSTEMS JOURNAL, the IEEE ACCESS, the *Computer Networks* (Elsevier), and the *Microprocessors and Microsystems* (Elsevier). He is a Managing Editor of the *International Journal of Autonomous and Adaptive Communications Systems* and the *International Journal of Arts and Technology*. He is a Senior Member of the CCF and the CMES.

JIAXIANG LIN received the Ph.D. degree in communication and information system from Fuzhou University, China, in 2010. He is currently a Lecturer with the College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou, China. He has hosted two national, provincial, and ministerial level research projects and authored over 20 referred journal and conference papers. His research interests include spatial data mining, visualization, parallel and distributed computing, and cloud computing.

SHIYONG WANG was born in Huoqiu, China, in 1981. He received the B.S. and Ph.D. degrees in mechanical and electrical engineering from the South China University of Technology, Guangzhou, China, in 2010. Since 2010, he has been a Lecturer with the Mechanical and Electrical Engineering Department, South China University of Technology. He has authored over ten articles and holds four patents. His research interests include motion control, robotics, and embedded control system. He was a recipient of the First Prize for Science and Technology Development of Guangdong Province in 2009.

DI LI is currently a Professor with the School of Mechanical and Automotive Engineering, South China University of Technology, China. She has directed over 50 research projects, including the National Natural Science Foundation of China. Thus far, she has authored/co-authored over 180 scientific papers. Her research interests include embedded systems, computer vision, and cyber-physical systems.

SEUNGMIN RHO received the Ph.D. degree in computer science from Ajou University, South Korea, in 2008. From 2008 to 2009, he was a Post-Doctoral Research Fellow with the Computer Music Laboratory, School of Computer Science, Carnegie Mellon University. He is currently a Faculty Member with the Department of Media Software, Sungkyul University. His current research interests include database, big data analysis, music retrieval, multimedia systems, machine learning, knowledge management, and computational intelligence.

CHANGCAI YANG received the M.S. degree in control theory and control engineering from China Three Gorges University, China, in 2008, and the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology (HUST), China, in 2012. He held a post-doctoral position with HUST, China, from 2012 to 2014. He is currently a Lecturer and the M.S. Supervisor of Computer Science and its Application with Fujian Agriculture and Forestry University, Fuzhou, China. His research interests include image matching, image segmentation, and 2-D–3-D pose estimation. He has authored over 20 papers.