

7094 IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 16, NO. 11, NOVEMBER 2020

Distributed Blockchain-Based Trusted Multidomain Collaboration for Mobile Edge Computing in 5G and Beyond

Hui Yang, Yongshen Liang, Jiaqi Yuan, Qiuyan Yao, Ao Yu, and Jie Zhang



***Abstract*—Mobile edge computing (MEC) sinks comput-ing power to the edge of networks and integrates mobile access networks and Internet services in 5G and beyond. With the continuous development of services, privacy pro-tection is extremely important in a heterogeneous MEC system for multiserver collaboration. However, most of the existing schemes only consider the privacy of users or services other than the privacy of network topology. For the purpose of topology privacy protection, this article em-ploys blockchain to construct heterogeneous MEC systems and adopts accommodative bloom filter as a carrier for multidomain collaborative routing consensus without ex-posing topology privacy. Blockchain is used to implement multiplex mutual trust networking and collaborative routing verification through the membership service and consen-sus mechanism. Experiments are conducted to evaluate the feasibility and performances of our scheme. The results indicate that the proposed scheme can highly improve the credibility and efficiency of MEC collaboration.**

***Index Terms*—Blockchain, Internet of Things (IoT), mo-bile edge computing (MEC), network security, 5G and be-yond.**

1. INTRODUCTION

HE ERA of intelligent interconnection of everything is **T**accelerating the arrival with the rapid evolution of Internetof things (IoT) and 5G systems. It is estimated that by 2020, more than 24 billion devices will need to be connected to communication systems [1]. The increasing number of devices is

Manuscript received September 9, 2019; revised December 17, 2019; accepted January 1, 2020. Date of publication January 7, 2020; date of current version July 29, 2020. This work was supported in part by the National Natural Science Foundation of China under Project 61871056, in part by the Young Elite Scientists Sponsorship Program by CAST under Grant 2018QNRC001, in part by the Fundamental Research Funds for the Central Universities under Grant 2018XKJC06 and Grant 2019PTB-009, in part by the ZTE Research Fund, in part by the Fund of SKL of IPOC, Beijing University of Posts and Telecommunications, under Grant IPOC2018A001 and Grant IPOC2019ZT01, in part by the Key Laboratory Fund under Grant 6142411182112, Grant 614210419042, Grant 61400040503, and Grant CEPNT-2017KF-04, and in part by Beijing Natural Science Foundation under Grant 4202050. Paper no. TII-19-4158. *(Corresponding author: Hui Yang.)*

The authors are with the State Key Laboratory of Information Pho-tonics and Optical Communications, Beijing University of Posts and [Telecommunications, Beijing 100876, China (e-mail: yanghui@bupt.](mailto:yanghui@bupt.edu.cn) [edu.cn; 378014739@qq.com; 3095384865@qq.com; yqy89716@bupt.](mailto:yqy89716@bupt.edu.cn) [edu.cn;](mailto:yqy89716@bupt.edu.cn) [yuaoupc@163.com;](mailto:yuaoupc@163.com) [lgr24@bupt.edu.cn)](mailto:lgr24@bupt.edu.cn)[.](mailto:yqy89716@bupt.edu.cn)

Color versions of one or more of the figures in this article are available online at [http://ieeexplore.ieee.org.](http://ieeexplore.ieee.org)

Digital Object Identifier 10.1109/TII.2020.2964563

accompanied by the high growth of data. According to the Cisco global cloud index [2], the annual global cloud computing traffic will reach 19.5 ZB by the end of 2021, up from 6.0 ZB per year in 2016, which poses a huge challenge to the computing power of 5G systems.

In the face of the explosive growth of IoT and 5G services, such as high-definition video, automatic drive, and augmented reality, etc., the cloud computing model gradually shows their shortcomings in the real-time process, network constraints, re-source overhead, and privacy protection. To enable massive and complex computing services, heterogeneous mobile edge computing (MEC) systems were proposed to carry out the emerging services [6]–[9]. Heterogeneous MEC is a distributed computing platform that integrates multiple MEC servers, cloud servers, computing, storage, and application competence and provides large-scale edge intelligent services on the edge side of the mobile network near the devices or data sources [4]. Due to the shortening of transmission links, edge computing can respond to service requests quickly and efficiently on the data side. In addition, edge computing can also provide basic service offline, which reduces the dependence of services on the network [5], [6].

In heterogeneous MEC systems, in addition to the need for a single server to quickly complete computing tasks on the edge side, multiple MEC servers need to collaborate to integrate the resources they own to complete large-scale computing tasks or data migration [12]. However, multiserver collaboration typi-cally involves multiple servers belonging to different domains whose trust is isolated from each other since they are managed by different organizations. Therefore, there are trust and security issues when MEC servers of different domains need to collabo-rate with other servers [10]–[12].

In practice, in front of us is mainly a contradiction between trust and privacy protection of multiple domains in MEC sys-tems. Each domain usually has a software-defined network (SDN) controller for centralized software-defined control of the underlying network and routing establishment. MEC servers communicate with other servers through the cross-domain rout-ing delivered by the SDN controller. To ensure that the cross-domain routing of multiple servers in the MEC task is trusted, SDN controllers need to obtain topology information of other domains to verify the legitimacy of the routing. However, the topology information across different domains should be usually confidential to each other, including node connection

1551-3203 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:22:27 UTC from IEEE Xplore. Restrictions apply.

|  |  |
| --- | --- |
| YANG *et al.*: DISTRIBUTED BLOCKCHAIN-BASED TRUSTED MULTIDOMAIN COLLABORATION FOR MOBILE EDGE COMPUTING | 7095 |
|  |  |

relationships, traffic information on links, resource utilization in the current network, etc. Once the privacy leakage occurs, criminals can obtain the complete topology data to analyze the traffic status to track user behavior and steal data by disguising as computing nodes, etc., which will do great harm to the MEC system.

Blockchain [13], [14], [25] is a secure distributed architec-ture to support the credible distributed communications and essentially possesses key characteristics of decentralization, distributed consistency, consensus, etc., which can effectively resolve the contradiction mentioned above. The consistency of that shared topology information is guaranteed by the blockchain multiparty consensus, while any malicious behavior of faulty nodes below the threshold will not affect the correct operation of the consensus. On the other hand, blockchain, owning complete data encryption technology, can resolve the privacy protection issue in MEC. Therefore, we use blockchain, with the above-mentioned properties, as the carrier of data sharing between MEC servers and make it the basis for trusted collaboration of multidomain MEC networks.

*Contribution:* In this article, we propose a novel distributedblockchain-based trusted MEC collaboration (BlockTC) ar-chitecture, which can accomplish the credible cross-domain collaboration among multiple MEC server verification. In addition, for server collaboration and privacy protection in MEC, we introduce two distributed multiserver collabora-tion routing verification (CRV) schemes, which have no need to disclose the private information of each domain while MEC servers collaborate. The overall feasibility and effi-ciency of the proposed architecture with CRV schemes are experimentally verified on the MEC-embedded blockchain testbed.

*Organization:* The rest of this article is structured as follows.Section II discusses the related work on the privacy protection about heterogeneous MEC and combination with blockchain in MEC of distributed networks. Section III presents the proposed BlockTC architecture of this article. Section IV describes how the architecture protects the topology privacy in MEC systems. Section VI shows the evaluation of the proposed architecture based on different performance metrics. Section VI concludes this article.

1. RELATED WORK

Similar to cloud computing, MEC actually faces many secu-rity problems and challenges in recent research. For instance, the interaction of heterogeneous edge nodes and service migration across global and local scales creates the possibility of malicious attacks. Personal privacy data contained in computing data, such as personal clinical data and financial information, can be easily stolen by attackers in these scenarios [15], [16]. To resolve these problems, authors in [17] abstract the processing and generation of user privacy data into linear programming (LP) computing rather than ordinary circuit representation in the data preprocessing phase. An efficient privacy protection problem conversion technology is developed, which allows customers to convert the original LP into random LP while protecting

sensitive input/output information. LDPMiner [18], a two-phase mechanism for acquiring heavy hitters with local different pri-vacy, uses a part of the privacy budget to collect a candidate set of heavy hitter and then, concentrates the remaining budget on the second stage to optimize the candidate set, which is more efficient than directly extracting the heavy hitters from the entire dataset.

As an emerging distributed network architecture, blockchain has drawn a great deal of attention in MEC systems. Xiong *et al.*, in [19] research the edge computing resource manage-ment based on optimal pricing to support mobile blockchain applications, which can be offloaded to edge computing servers, which adopts a two-phase game to maximize the benefit of edge computing servers and the individual profit of users. In addition, many researchers have also introduced blockchains in MEC systems for data privacy protection. In [20]–[23], blockchain is introduced into distributed fault-tolerant control and anonymous access of IoT, which effectively protects the edge privacy of edge computing nodes and IoT devices. Authors in [26] formulate the MEC task offload and user privacy protection as a joint optimization issue to minimize MEC system offload costs and maximize the privacy protection of all blockchain users. In [27], the blockchain and ToR (The Onion Router) technology have been introduced to make MEC data immutable, anonymous, secure, and transparent to relevant communities which realize the privacy protection of MEC data in the life cycle of data generation and sharing.

Each of the above mentioned schemes has its own advantages and disadvantages, but the privacy of network topology in MEC systems is not taken into account yet. To address this problem, we design a distributed architecture of interdomain topology privacy protection in MEC.

1. SYSTEM DESIGN AND FUNCTION MODULE

By introducing a blockchain into the heterogeneous MEC system, we propose a BlockTC architecture which is illustrated in Fig. 1 to solve the problem of routing trust and privacy protection. In a multidomain MEC network, routing services are mainly classified into intradomain routing and cross-domain routing. Intradomain routing can be considered safe because it involves only nodes and links within the domain. Therefore, BlockTC mainly provides trusted routing solutions for cross-domain routing in this article.

BlockTC focuses on the multidomain MEC collaboration on the edge side to achieve multiserver trusted routing and privacy protection, owning the following characteristics.

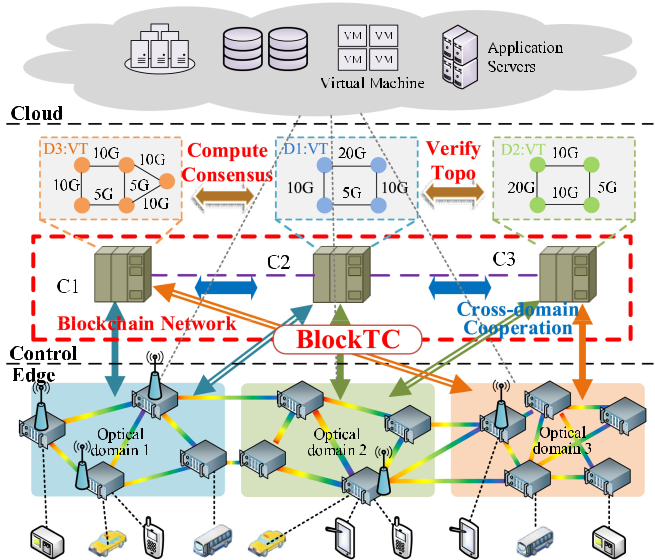
*A. Backup Dual Link*

The mapping between the controllers and MEC servers is the backup dual link for maintenance and verification. The intent of this is to share the virtual topology of a certain domain with a verification controller (not the main controller of the domain) as the basis for consensus verification of MEC col-laboration routing without exposing privacy across the network. Virtual topology is composed of network links and resource

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:22:27 UTC from IEEE Xplore. Restrictions apply.

7096 IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 16, NO. 11, NOVEMBER 2020

Fig. 1. Architecture of BlockTC for MEC.



weights, without physical and identity information of topology components.

*B. Distributed Ledger*

Each controller loads a blockchain ledger, whose data are initialized by the backup dual link and updated by consensus. Distributed ledgers have a characteristic of strong consistency. The physical and virtual topology information status stored in ledgers can synchronize in real time, also providing a data basis for consensus.

*C. Collaboration Routing Consensus*

According to the multidimensional topology data stored in blockchain ledgers, controllers can independently complete the MEC collaboration routing of its domain which is then output to the blockchain network for verification and also verify the routing received from other controllers, thus realizing an MEC collaboration routing consensus.

These characteristics provide a feasible BlockTC architecture for trusted MEC collaboration. Topology data stored in the con-troller’s database can be divided into two categories—physical topology data of its domain and virtual topology data of its verification domain. For example, as shown in Fig. 1, controller C1 manages its own domain 1 as a main controller to protect its topology information privacy, while maintaining the virtual topology of domain 3 as a verification controller to verify whether the collaboration routing of C3 is correct and realize col-laboration routing consensus. Among them, blockchain ledgers store the topology resource update of each routing result to maintain the correctness of the virtual topology data. Therefore, the BlockTC can not only prevent the privacy of its topology traffic from being exposed to other domains but also provide evidence of cross-domain routing verification.

In the edge side, MEC servers are deployed along with wireless access points (APs) such as base stations which act as source or destination nodes of routing when a cross-domain MEC task arrives. To ensure the QoS of cross-domain MEC services, the data interaction between MEC servers will use the optical link between APs to transmit, through optical and wireless integration technology to meet the demands of low latency and multidimensional resource utilization.

In the cloud side, the role of cloud servers is consistent with the existing MEC solutions. The wireless APs enable the access to remote cloud servers through backhaul links, helping the MEC server to further offload some computing tasks to large-scale cloud data centers, which will be used as some large-scale applications, such as big data analytics and machine learning, to improve the QoS for edge computing and cloud computing.

In the control plane, controllers of BlockTC are composed of several SDN controllers for centrally controlling corresponding physical resources of different dimensions, such as radio, optical fiber, and MEC servers.

Radio resources, including radio channels and power, are managed by radio controllers to provide access services to virtually all MEC devices. Radio controllers perform the radio channel allocation function to allocate the radio channel it owns, and adopt the OpenFlow control function to implement channel modification.

Optical resources mainly refer to fiber frequency slots which can load computing data for interserver transmission and data backhaul to cloud servers in distributed edge computing. Optical controllers perform the network virtualization to virtualize the optical resources for centralized control and furthermore execute the path computing and optical resource allocation while data transmission request arrives.

MEC server resources refer to the application resources of MEC servers, including memory, CPU, I/O, etc., which are the most critical resources for edge computing. MEC servers cen-trally control all MEC tasks and allocate application resources of the MEC server to those tasks.

In order to implement the SDN protocol on global resources, different controllers maintain different types of resources and interwork with each other through a unified OpenFlow agent to finally achieve the global resource control. More importantly, multidimensional resource collaborative control enables the SDN control plane to simultaneously allocate all resources of the data plane to achieve global optimality, which greatly enhances the flexibility and availability of MEC systems. For more infor-mation on the multidimensional resource collaborative control used in this article, please refer to [28]–[30].

In the following content, we abstract the multidimensional controllers into BlockTC controllers and summarize their func-tional modules as shown in Fig. 2. The trusted control mod-ule can accomplish the membership trusty of the blockchain network and the verification trusty of MEC collaboration routing. Using a membership trusty module originating from blockchain, multiple controllers can acquire the credible identity of blockchain networks and be identified by each other in the distributed consensus. The verification trusty module verifies the intradomain routing of other controllers to ensure that the

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:22:27 UTC from IEEE Xplore. Restrictions apply.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| YANG *et al.*: DISTRIBUTED BLOCKCHAIN-BASED TRUSTED MULTIDOMAIN COLLABORATION FOR MOBILE EDGE COMPUTING | | | | | | | | | | | | | | | | | | | 7097 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

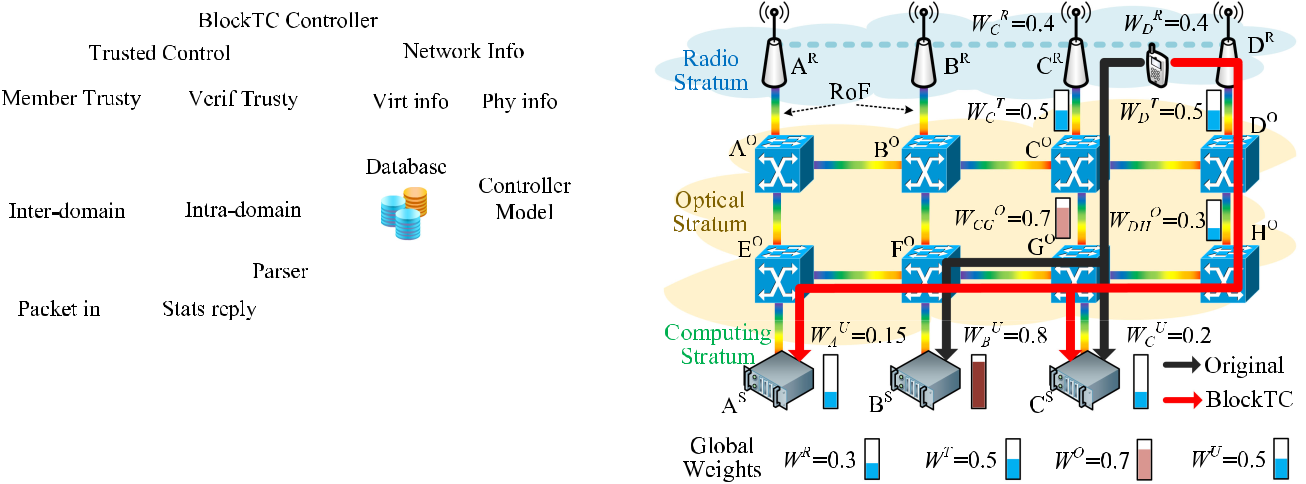


Fig. 2. Functional model of BlockTC.

results are consistent with the virtual topology information. Network information of the controller’s own domain is collected in the physical information module, while other domain’s virtual topology should be maintained in a virtual information module to prepare for routing consensus verification. Multidomain MEC collaboration routing is performed in the path compute element including interdomain and intradomain processes, considering the MEC servers load and multimode communication resources. Parser module performs the conventional functions of SDN con-trollers, including continuous spectrum assignment, provisions the path using OpenFlow protocol, and so on.

IV. IMPLEMENTATION OF BLOCKCHAIN-BASED TRUSTED COLLABORATION FOR MEC

This section elaborates on the implementation of the proposed scheme from two aspects, including routing algorithm and rout-ing verification.

*A. MEC Collaboration Routing Algorithm*

Heterogeneous MEC systems usually involve multidimen-sional resource couplings. In traditional schemes, the radio path, as well as optical path computing and spectrum alloca-tion, will be divided into several parts independently, which make it easy to obtain a local optimal solution, rather than a global optimal one. Local optimal solutions are not conducive to network persistent operation in terms of resource utilization or communication delays. Allocating resources to new services based on a local optimal solution may lead to the traffic block, thus affecting network operation. Therefore, we propose a novel MEC collaboration routing algorithm for BlockTC to support MEC collaboration using coupled path with radio, optical, and computing resources.

Fig. 3 shows an instance BlockTC network in an MEC system which is composed of four radio APs, eight optical switching nodes, and three MEC servers. According to the principle of shortest path and proximity, in original schemes, distributed MEC services initiated by the user equipment (UE) will of-fload to servers *BS* and *CS* through the black path in Fig. 3. Server *BS* and link *lCG* are heavily loaded, which may cause other services to be blocked due to resource exhaustion, thus reducing the QoS. Using the proposed algorithm, controllers

Fig. 3. Illustration of auxiliary graph for BlockTC scheme.

compute the weights for each link and server, and select the path with the minimum sum of weights (such as the red line in Fig. 3), so as to reduce the burden of heavy load units and balance global resource load. In this way, the global optimal solution can be obtained and implemented to effectively improve resource utilization and reduce blocking.

In order to facilitate the storage of topology and com-plete the abovementioned weight calculation, we use an aux-iliary graph to represent the physical topology in BlockTC, as *G* (*N,* *N* *,* *L,* *L* *,* *S,* *S* *,* *C*). *N* = *{n*1*, n*2*, . . . , nn}* and *N* = *{n*1*, n*2*, . . . , nn* *}* represent the set of optical switching nodes and radio access points, respectively. *L* =

*{l*1*, l*2*, . . . , ln}* and *L* = *{l*1*, l*2*, . . . , ln}* denote the set ofbidirectional optical paths or radio channels between the nodes

in ***N*** and *N* . *S* = *{s*1*, s*2*, . . . , sn}* and *S* = *{s*1*, s*2*, . . . , sn}* are the set of optical spectrum and the radio frequency on each

link of ***L*** and ***L*’**. Also, ***C*** indicates the set of MEC servers. When an MEC request arrives, jointly considering the weights

of radio dimension, radio over fiber (RoF) dimension, fiber spec-trum dimension, and computing dimension, controllers compute the globally optimal servers and paths for this request. Specifi-cally, the computation of weight can be divided into four aspects. Each resource weight has a priority parameter, and the sum of all priority parameters is always 1 to satisfy the normal-ization. Within this range, administrators can adjust the level of various priority parameters to make the resource scheduling more flexible. As shown in Fig. 3, the optical resource weight is the highest one in the current global resource weights. In order to make the resource allocation more balanced, the priority parameter corresponding to the optical resource weight can be selected as a higher level. Conversely, the priority parameters corresponding to other resources will be adjusted to a lower level. This will make a larger gap between the weight of light resources and that of other resources, so that more consideration will be given to optical resources in resource allocation until the priority parameter of optical resources is reduced after it becomes idler.

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:22:27 UTC from IEEE Xplore. Restrictions apply.

7098 IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 16, NO. 11, NOVEMBER 2020

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| In order to represent the workload capacity of radio edge | | | | | | | |  |  |  |  |  |  |  |  |
| **Algorithm 1:** Routing Algorithm in BlockTC. | | | |  |  |  |  |
| between UE and APs, we use the wireless spectral utilization, | | | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 1: **Input:** *G (V, V’, L, L’, F, F’, A), T (*src*, b, c)* | | | |  |  |  |  |
| and the ratio of distance to power on radio links to compute | | | | | | | |  |  |  |  |
| 2: | **Output:** Provisioning path | | |  |  |  |  |
| the radio edge weight *WnR* for the MEC request, as in (1). *RFn* | | | | | | | |  |  |  |  |
| and *RF*0 are the occupied and total frequency slots on the radio | | | | | | | | 3: | Construct auxiliary graph and calculate weights *W* *R*, | | | | | |  |
|  |  |  |  |  | *n* | |  |
|  | *W T* , *W O* and *W U* of relevant antennas, links, and | | | | |  |  |
| edge, while *Dn* and *Pn* present the distance and power between | | | | | | | |  |  |  |
|  | *n* | *n* | *n* |  |  |  |  |
|  | servers; |  |  |  |  |  |  |
| UE and AP. Meanwhile, *α* and *β* are the priority parameters of | | | | | | | |  |  |  |  |  |  |  |
| 4: | Select K candidate servers with minimum *WnU* , which | | | | | |  |
|  |  |  |  |  | *R* |  |  |  |
| the radio resource weight *Wn* , which can be adjusted to change | | | | | | | |  |  | *Ks* |  |  |  |  |  |
| the priority of radio resources during resource allocation. How- | | | | | | | | 5: | satisfy | =1 *UnC > c*; | |  |  |  |  |
| ever, since distance and power belong to different magnitudes, | | | | | | | | **for** *s*=1,2, …,*K* **do** | | |  |  |  |  |
| 6: | Run Dijkstra algorithm from src to server *As* with | | | | | |  |
| *Dn/Pn* is not a normalized indicator like *RFn /RF*0. In order | | | | | | | |  |
|  | the minimum sum of relevant edge weight; | | |  |  |  |  |
| for *Dn/Pn* and *RFn* */RF*0 to have the same proportion in *WnR*, | | | | | | | | 7: |  |  |  |  |
| *β* is also used to unify them to the same standard | | | | | | |  | **if** no path is found **then** | | |  |  |  |  |
|  | 8: | Block the task request; | | |  |  |  |  |
|  | *WnR* = *α · RFn/RF*0+ *β · Dn/Pn.* | | | | | | (1) |  |  |  |  |
|  | 9: | **else** |  |  |  |  |  |  |
| RoF is a technology that modulates data from radio into light | | | | | | | | 10: | Record route *Rs* as one of candidate route set ***R***; | | | | | |  |
| 11: | **end if** | |  |  |  |  |  |
| and transmits it in optical fiber. Its resource weight usually | | | | | | | |  |  |  |  |  |
| 12: | **end for** |  |  |  |  |  |  |
| considers the bit rate and radio frequency. Thus, the radio edge | | | | | | | |  |  |  |  |  |  |
| 13: | Perform *routing verification* | | |  |  |  |  |
| weight *WnT* can indicate the modulation cost as in (2). Here, *Bn* | | | | | | | |  |  |  |  |
| 14: | **if** success **then** | |  |  |  |  |  |
| and *Fn* denote the bit rate and radio frequency of current radio | | | | | | | |  |  |  |  |  |
| 15: | route ***R*** according to the final path; | | |  |  |  |  |
| signals, respectively. Also, *γ* is used to normalize the RoF weight | | | | | | | |  |  |  |  |
| 16: | **if** the path contains RoF or optical edges **then** | | |  |  |  |  |
| and adjust the priority of RoF resources | | | | | |  |  |  |  |  |  |
|  |  | 17: | Establish a new spectrum path based on *W* *R*, *W* *O* | | | | | |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | *n* |  |  | *n* |  |
|  |  |  |  | *T* | = *γ* *·* *Bn/Fn.* |  | (2) |  | of the selected RoF and optical edges; | | |  |  |  |  |
|  |  |  | *Wn* | |  |  |  |  |  |  |
|  |  |  |  | 18: | **end if** | |  |  |  |  |  |
| In the optical dimension, we mainly consider spectrum re- | | | | | | | |  |  |  |  |  |
| 19: | **if** the path contains radio edges **then** | | |  |  |  |  |
| sources, which are used to carry MEC service data in optical | | | | | | | |  |  |  |  |
| 20: | Route the request in radio network based on edge | | | | | |  |
| fiber. In the process of evaluating the accommodation capacity | | | | | | | |  |
|  | weight *W* *R*; | | |  |  |  |  |
| of the optical spectrum, it is necessary to consider the constraints | | | | | | | |  |  | *n* |  |  |  |  |  |
| 21: | **end if** | |  |  |  |  |  |
| of the optical spectrum continuity. In order to carry a new service, | | | | | | | |  |  |  |  |  |
| 22: | Update the network status and weights *W* *R*, *W* *T* | | | | | , |  |
| the optical spectrum must reserve *b* + *B* continuous available | | | | | | | |  | *W O*, and *W U* | | *n* | *n* | |  |  |
|  | ; |  |  |  |  |
| bandwidth, in which *b* and *B* represent the bandwidth required | | | | | | | |  | *n* | *n* |  |  |  |  |  |
| 23: | **else** |  |  |  |  |  |  |
| by the new service and the guard bandwidth. Specifically, *cs* and | | | | | | | |  |  |  |  |  |  |
| 24: | Fail to route and report errors; | | |  |  |  |  |
| *bi* describe the count of possible spectrum allocation cases and | | | | | | | |  |  |  |  |
| 25: | **end if** |  |  |  |  |  |  |
| the bandwidth required by the *i*th case. Therefore, the average | | | | | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| bandwidth *bns* of *s*th subcarrier on *ln* can be described as in (3). | | | | | | | |  |  |  |  |  |  |  |  |
| To evaluate the resource weight of the lightpath, we also need to | | | | | | | | standard, which satisfy *kC* + *kR* + *kI* as a fixed value, as the | | | | | | |  |
| consider the change of resource occupancy of adjacent subcarri- | | | | | | | |  |
| priority parameter of MEC servers. Therefore, we can use (5) to | | | | | | |  |
| ers on *ln*, which is defined as *vn*. In summary, the optical resource | | | | | | | |  |
| describe the resource weight *W* *U* of each MEC server. In Fig. 3, | | | | | | |  |
| weight *W* *O* is presented to measure the spectrum utilization as | | | | | | | | *U* |  | *U* | *n* |  | *S* |  |  |
|  |  |  |  |  |
| *n* |  |  |  |  |  |  |  | *WA* | = 0*.*15 and *WB* = 0*.*8, which means that server *B* | | | |  | is in |  |
| in (4). Identically, *μ* is used to normalize the optical weight and | | | | | | | |  |  |
| a high load status. According to load balance, the MEC task | | | | | | |  |
| adjust the priority of optical resources. As shown in Fig. 3, *WCGO* | | | | | | | |  |
| should be offloaded to server *AS* | | | |  |  |  |  |
| is higher than *W* *O* | | | . Therefore, from the perspective of optical | | | | |  |  |  |  |  |  |  |  |
|  |  | *DH* |  |  |  |  |  |  | *WnU* = *kC · UnC* + *kR · UnR* + *kI · UnI.* | | |  |  |  |  |
| resources alone, the routing path represented by the black line | | | | | | | |  |  |  | (5) |  |
| is shorter than the red one, but its resource load is higher, which | | | | | | | | When a new MEC collaboration task arrives, request *T (*src*,* | | | | | | |  |
| will easily lead to traffic block | | | | | |  |  |  |
|  |  | *b, r)* arrives at the network, in which src, *b*, and *r*, respectively, | | | | | | |  |
|  |  | *cs* |  |  |  |  |  |  |
|  |  |  |  |  |  |  | represent the source of the MEC service request, as well as | | | | | | |  |
| *bs* | = | *b* | */c* |  |  |  | (3) |  |
|  |  |  | the required bandwidth, and computing resources, and then the | | | | | | |  |
| *n* |  | *i* |  | *s* |  |  |  |  |
|  |  | *i*=1 |  |  |  |  |  | controller establishes an auxiliary graph and computes weights | | | | | | |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | *F* | *F* | *cs* |  | of the auxiliary graph according to the current network state | | | | | | |  |
| *WnO* = *μ · vn* | | |  |  | *bns* = *μ · vn ·* | *bi* | *cc.* (4) |  |
|  |  | for routing computing. The pseudocode of the abovementioned | | | | | | |  |
|  |  |  |  | *s*=1 | *s*=1 *k*=1 | |  | routing algorithm and the subsequent path establishment is given | | | | | | |  |
| In the MEC servers’ dimension, in order to represent the | | | | | | | | in Algorithm 1. | | |  |  |  |  |  |
| server resource load status, several factors need to be considered, | | | | | | | | The MEC collaboration routing algorithm in BlockTC com- | | | | | | |  |
| including CPU utilization *UC*, RAM utilization *UR*, and I/O | | | | | | | | prehensively considers the resource weights of multiple dimen- | | | | | | |  |
| scheduling ratio *UI* . Besides, we take three adjustable param- | | | | | | | | sions and provides a global optimal solution. In this procedure, | | | | | | |  |
| eters *kC*, *kR*, and *kI* to unify those resources with the same | | | | | | | | each controller performs the routing computing independently | | | | | | |  |

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:22:27 UTC from IEEE Xplore. Restrictions apply.

|  |  |
| --- | --- |
| YANG *et al.*: DISTRIBUTED BLOCKCHAIN-BASED TRUSTED MULTIDOMAIN COLLABORATION FOR MOBILE EDGE COMPUTING | 7099 |
|  |  |

Fig. 4. Interactive procedure of ND-CRV.

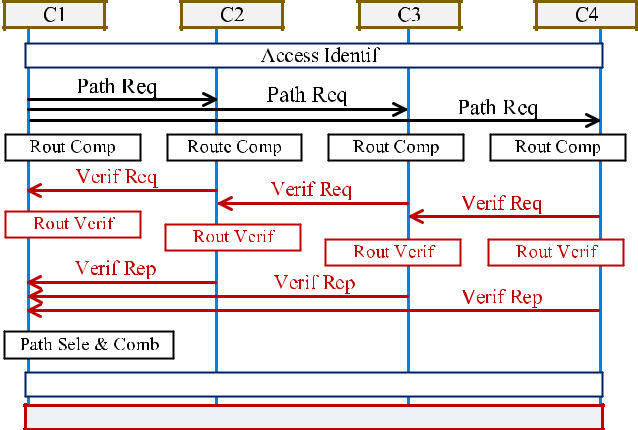
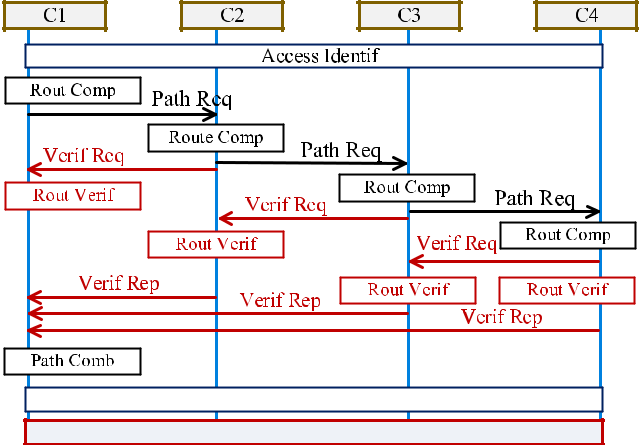


Fig. 5. Interactive procedure of CD-CRV.

and obtains the candidate routing sets. Subsequently, all con-trollers perform multidomain routing verification based on can-didate routing sets, and only if the verification succeeds will path combination and establishment be carried out.

Meanwhile, the MEC servers will update links and the corre-sponding weights described above to the verification controllers for collaboration routing verification, through the backup dual link. Controllers collect the virtual topology element *E* (*li, Wi*) into virtual topology set *E* = *{E*1*, E*2*, . . . , En}* at this stage.

*B. MEC Collaboration Routing Verification*

Based on the BlockTC architecture and candidate routing sets calculated by each domain, we propose two blockchain-based routing verification schemes for MEC collaboration to imple-ment the secure routing without any exposure of private data. Network-driven CRV (ND-CRV) and cloud-driven CRV (CD-CRV) schemes are presented in Figs. 4 and 5. Controllers con-stitute the consensus group of blockchain network through strict membership access and management mechanism in blockchain.

Fig. 4 depicts the interactive procedure of ND-CRV in BlockTC. ND-CRV refers to the PCE-based interdomain path

computing, in the case that the controller undertakes all the tasks of the MEC servers. Through the blockchain-based trusted authentication, each controller generates and maintains a cred-ible access identity to perform the distributed consensus in the blockchain network. First, we assume that a cross-domain MEC request *T* (src, *b*, *c*) arrives at controller 1 (C1) through domain

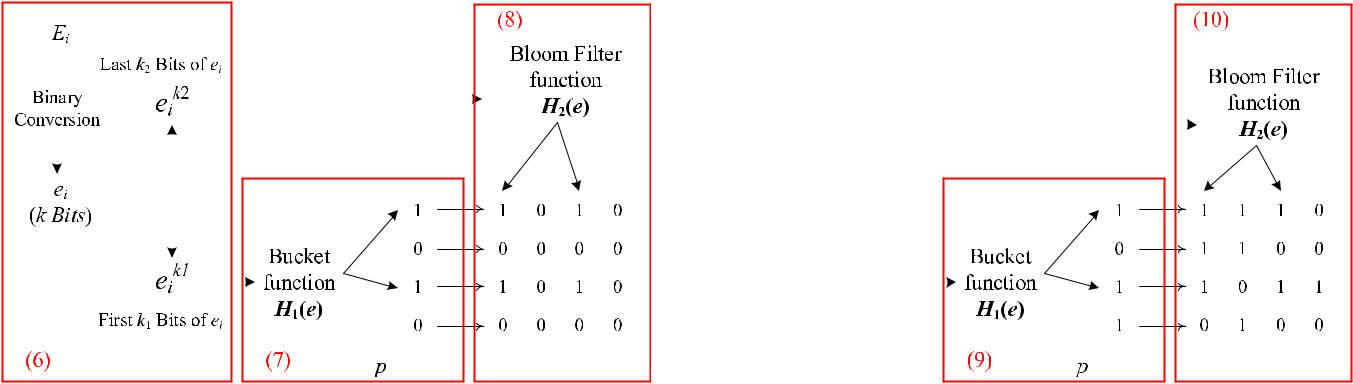
1. According to the request, C1 computes the intradomain path within domain 1 and selects the best one with the minimum sum of weights, while sending a new request to the controller of the subsequent domain which owns the virtual topology of domain
2. The new request *T*1 = {*T*, *BF*1}, in which *T* and *BF*1 are the original request and the bloom filter containing C1 routing results, respectively. After intradomain path calculation and selection, the subsequent controller C2 sends the verified request with a new bloom filter including the intradomain path of domain
3. to the next controller that owns its virtual information to make sure the route trust. An intradomain path can be identified using virtual info with bloom filters, and then, the credible path result is returned to SC1. This subprocedure will be repeated until the request reaches the destination domain. Finally, C1 receives all the verified paths from multidomain controllers and combines those paths into a trusted multidomain routing. The multiserver MEC task is accommodated with the routing provisioning among controllers. In ND-CRV, each controller selects the optimal intradomain path first and then, verifies the routes from other domains.

Different from ND-CRV, the CD-CRV scheme is suitable for situations where controllers are insensitive to the feature of MEC servers. In such a situation, the operator predetermines a sparse route between domains and sends corresponding MEC task requests to each controller simultaneously. Therefore, each controller does not need to consider the interdomain routing, and can simultaneously calculate and verify the intradomain routing, thus, speeding up the routing procedure. After receiv-ing the request, controllers compute the total weights of each candidate route between any of the entrances of its domain to that of the subsequent domain. Afterward, they send the routing verified requests with multiple candidate paths to subsequent controllers to accomplish the routing verification. The approach of routing verification in CD-CRV is consistent with ND-CRV scheme, but it changes the verification from a single path to multiple paths. By comparing the weight combination of each route, controllers can select a cost-efficient route from the cred-ible paths as the final routing. In CD-MCR, each controller implements the verification first and then performs the path selection.

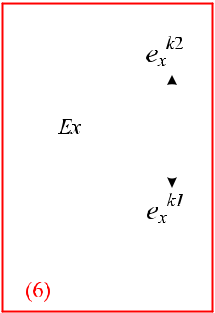
In these two verification processes, BlockTC adopts the blockchain networking mode of the consortium chain. Each practical SDN controller stores the corresponding resource weights of its verification domain. When multidomain routing verification is required, SDN controllers would endorse the routing result to confirm its correctness. On the other hand, these controllers will form a logical entity (or we call it the BlockTC controller hereinbefore) to perform consensus and sort the transactions into blocks. In summary, each SDN controller will be able to perform routing verification efficiently through this blockchain.

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:22:27 UTC from IEEE Xplore. Restrictions apply.

7100 IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 16, NO. 11, NOVEMBER 2020



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Fig. 6. Detailed process of the insertion of BlockTC-enable ABF. | | | | | | | | | | | | | | | | | | | | | | | Fig. 7. Detailed process of the querying of BlockTC-enable ABF. | | | | | | | | | | | | | | | |  |  |
|  | Here, we describe the implementation of the verification pro- | | | | | | | | | | | | | | | | | | | | | | bucket in bucket level is divided into *k*2 parts and each hash | | | | | | | | | | | | | | | | | |
| cedure. In the routing verification procedure of BlockTC, after | | | | | | | | | | | | | | | | | | | | | | | function is used to update the corresponding part of the bloom | | | | | | | | | | | | | | | | | |
| obtaining the updated links from the underlying network, the | | | | | | | | | | | | | | | | | | | | | | | filter | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |
| verifier controller needs to determine whether each link element | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *Ei* (*li, Wi*), representing the resource weight *Wi* of link *li* of | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  | *cK*=2 | 1[*H*2 *eik*2 *→* *Pc*] *.* | | | | | | | | | | (8) | |
| these updated links is in the routing set provided by the prover | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  | *cK*=2 | 1*BFBj* [*Pc*] =1 | | | | | | | | | |  |  |
| controller. To ensure the high quality of MEC service processing, | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| we use an accommodative bloom filter (ABF), a kind of unidi- | | | | | | | | | | | | | | | | | | | | | | | In heterogeneous MEC scenarios, the scale of network topol- | | | | | | | | | | | | | | | | | |
| rectional accumulator [24], as the carrier for routing verification, | | | | | | | | | | | | | | | | | | | | | | | ogy is constantly changing, so that the amount of routing set data | | | | | | | | | | | | | | | | | |
| which not only reduces the size of data in the consensus, but also | | | | | | | | | | | | | | | | | | | | | | | may be so large that it possibly exceeds the security threshold of | | | | | | | | | | | | | | | | | |
| ensures that the third party cannot access routing information. | | | | | | | | | | | | | | | | | | | | | | | the bloom filter. In this case, we can append a slice to the bloom | | | | | | | | | | | | | | | | | |
| Especially, ABF can adjust the carrier volume according to the | | | | | | | | | | | | | | | | | | | | | | | filter to accommodate a larger amount of data. The size of such | | | | | | | | | | | | | | | | | |
| amount of data to meet the verification correctness requirement, | | | | | | | | | | | | | | | | | | | | | | | a new slice *s* is given as *s* = *q*/*K*2. | | | | | | | | | | | | | | | |  |  |
| which is well-adapted to the dynamic characteristics of the MEC | | | | | | | | | | | | | | | | | | | | | | | *2) Querying in BlockTC-Enable ABF:* The greatest advan- | | | | | | | | | | | | | | | | | |
| network. | | | | | | | | | | | | | | | | | | | | | | | tage of using ABF is the ability to quickly query whether an | | | | | | | | | | | | | | | | | |
|  | An ABF is composed of *p* buckets and each bucket involves | | | | | | | | | | | | | | | | | | | | | | element exists in a set or not. The verifier controller can verify | | | | | | | | | | | | | | | | | |
| an ordinary bloom filter of size *k*2 bits. Each link element *Ei* is | | | | | | | | | | | | | | | | | | | | | | | the correctness of the prover controller only by performing a | | | | | | | | | | | | | | | | | |
| first hash mapped to *k* bits (*k* = *k*1 + *k*2), where the first *k*1 bits | | | | | | | | | | | | | | | | | | | | | | | composite hash operation on the routing data and mapping it to | | | | | | | | | | | | | | | | | |
| are mapped to the bucket level and the second *k*2 bits are mapped | | | | | | | | | | | | | | | | | | | | | | | the bloom filter. The whole querying procedure can be divided | | | | | | | | | | | | | | | | | |
| to bloom filters of corresponding buckets. By inserting the link | | | | | | | | | | | | | | | | | | | | | | | into two steps, which is shown in Fig. 7. | | | | | | | | | | | | | | | |  |  |
| element of the routing result, the prover controller generates an | | | | | | | | | | | | | | | | | | | | | | | *Step I:* At this point, the ABF received from the prover | | | | | | | | | | | | | | | | | |
| ABF for link querying of the verifier controller. The detail of the | | | | | | | | | | | | | | | | | | | | | | | controller has loaded all the link elements in the routing result. | | | | | | | | | | | | | | | | | |
| insertion and querying in BlockTC is presented as Fig. 6. | | | | | | | | | | | | | | | | | | | | | | | First, as in the insertion procedure, each link element *Ex* updated | | | | | | | | | | | | | | | | | |



1. *Insertion in BlockTC-Enable ABF:* Each link element *Ei* by the underlying network through backup dual links need to be

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| is converted into *k* bits using a hash function and divided into | | | converted into *ex* of size *k* bits as in (6). After that, the first *k*1 |  |
| two parts *eik*1 and *eik*2 , of size *k*1 and *k*2 bits, respectively | |  | bits are mapped into buckets by *K*1 hash functions*jK*=11*H*1(*e*) to |  |
|  | *ek*1 |  | determine in which bloom filter the data is located. If the status |  |
| *Ei → ei →* | (6) | of the resulting buckets is not 1, it is judged that this element is |  |
| *eki*2 *.* |  |
|  | *i* |  | not in the routing set and the verification fails, otherwise it goes |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| In the bucket level, the first *k*1 bits *eik*1 are mapped into | | | | | | | to the next step | | | |  |  |  |  |  |  |  |  |
| the buckets using *K*1 hash functions *jK*=11*H*1(*e*). For each hash | | | | | | |  |  | *K*1 | | *k*1 | | *→ Buj*] |  |  |  |  |  |
| mapping, the corresponding bucket is set 1. In particular, if | | | | | | |  | *j*=1[*H*1 | | *ex* | | True |  | StepII *.* | (9) |  |
| a bucket has been set 1, the status of this bucket will remain | | | | | | |  |  |  | *Buj* | *j*=1 | *, Buj* ==1 | |  | *⇒* |  |  |  |
|  |  |  | *K*1 |  |  |  |  |
| *K*1 |  |  | *k*1 |  |  |  |  |  |  |  |  |  |  | False *⇒* fail | | |  |  |
| unchanged |  |  |  |  |  |  |  | *∀* | |  |  |  |  |  |  |  |  |  |
| *j*=1 | [*H*1 | *e* | *i* | *→* | *Bj*] |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | *.* | (7) | *Step* | *II:* After the first step is successful, the second step will | | | | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *K*1 |  | = 1 | |  |  |  | further verify the existence of elements. The last *k*2 bits of *ex* are | | | | | | | | | | |  |
| *j*=1*Bj* | |  |  |  |  |
|  |  |  |  |  |  |  | hashed in the corresponding bloom filter. In particular, when the | | | | | | | | | | |  |
| In the bloom filter level, the remaining *k*2 bits *eik*2 | | | | | | are hashed | bloom filters associated with selected buckets have an additional | | | | | | | | | | |  |
| into the bloom filter corresponding to each bucket using *K*2 hash | | | | | | | slice, all slices should be judged. The existence of this element is | | | | | | | | | | |  |
| functions *cK*=21*H*2(*e*). Each bloom filter related to an activated | | | | | | | recognized only when the status of all buckets and bloom filters | | | | | | | | | | |  |

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:22:27 UTC from IEEE Xplore. Restrictions apply.

|  |  |
| --- | --- |
| YANG *et al.*: DISTRIBUTED BLOCKCHAIN-BASED TRUSTED MULTIDOMAIN COLLABORATION FOR MOBILE EDGE COMPUTING | 7101 |
|  |  |

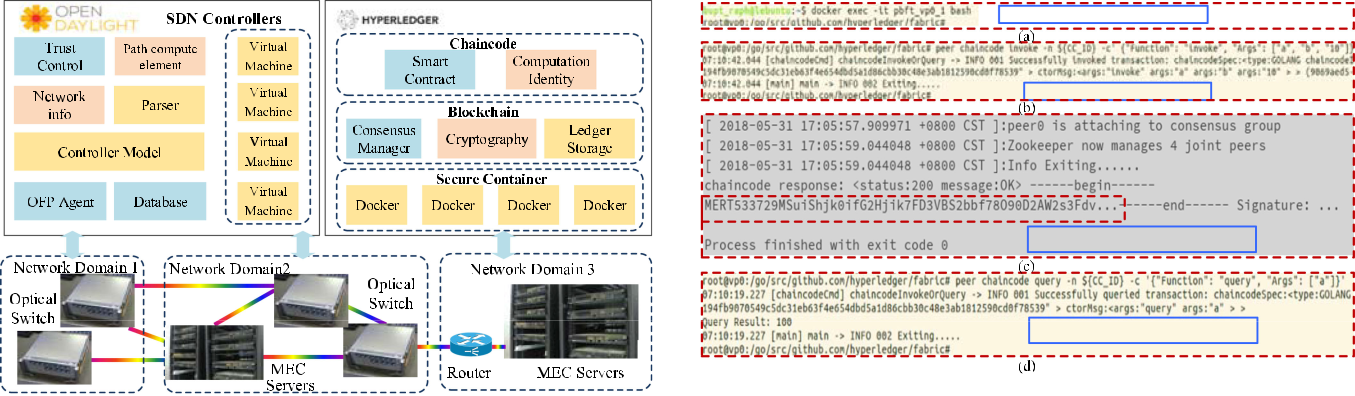


Fig. 9. Capture of access and routing verification for BlockTC.

Fig. 8. Experimental platform for BlockTC.

are 1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *cK*=2 | | 1[*H*2 | | *exk*2 | *→ Gc*] | True | *⇒* | success *.*(10) |  |
|  |  | *s* |  | *Gc* | *K*2 | *, Gc* ==1 |  |  |  |
|  | *∀* | *x* | |  | *c*=1 |  |  |  |  |  |
|  |  |  |  |  |  | False *⇒* fail | | |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

After both steps of verification are successful, the verifier controller determines that the verified elements exist in the routing set provided by the prover controller. As shown in Fig. 7, only when all the yellow squares are 1 is the querying considered successful.

1. SIMULATION AND RESULT ANALYSIS

In this section, we present the simulation and result analysis of the proposed BlockTC to demonstrate its network performance and verification performance. To estimate the feasibility of the proposed architecture, we deploy the BlockTC architecture based on our experimental platform, as shown in Fig. 8.

In terms of test data sources, we take real traffic from the State Key Laboratory datacenter in Beijing, China, as input in the proposed BlockTC. To test our system under different traffic load conditions, we select the traffic data of the most heavy-loaded 6 hours, including 15.7 million packets with a total size of 7.6 GB. By adjusting the traffic density, we simulate the performance of the proposed system under different traffic loads.

At the physical level, we deploy four ROADMs and two MEC servers to form a C-RoFN model [29], [31]. To evaluate the proposed system, the C-RoFN model is used to simulate the multidomain MEC offloading scenario. This multidomain model provides a multidimensional resource collaboration model from the radio access layer to the application layer, which is suitable for distributed network scenarios such as MEC and IoT. Among this model, in each MEC server, we deploy ten independently running Docker containers to simulate edge computing nodes in the test network for task offloading. By comparing the real traffic we intercepted with the traffic in an actual multidimensional resource hybrid network model [32], [33], we select to divide the above-mentioned C-RoFN model into three management domains to form a mesh network with 20 MEC computing nodes for simulation.

At the software level, we mainly deploy SDN controllers based on Opendaylight (ODL) Beryllium and a blockchain network based on Hyperledger Fabric V1.0. Among them, we develop MEC servers and SDN controllers in virtual machines running on several high-performance servers. Controllers based on ODL can complete the software-defined control functions, such as path computing, routing verification, flow table issuing, etc. In addition, within SDN controllers, we develop OpenFlow protocol agents to simulate other units and interconnect them in the proposed architecture to implement distributed MEC collaboration. In the Hyperledger-based blockchain network, various functions of blockchain, including identity authentica-tion, consensus, smart contract, and ledger management, are performed to accomplish the backup dual link, collaboration routing consensus, and membership management.

Based on the experimental platform, we have designed and verified experimentally the BlockTC for trusted collaborative route and routing verification in MEC for 5G systems. In the Hyperledger Fabric platform, we deploy the cross-domain col-laboration routing computing and verification schemes to the consensus nodes (controllers) in the form of smart contracts. By simulating MEC collaboration services, we get effective results on the platform. Fig. 9 shows the membership identification and routing verification results of BlockTC through the terminal capture in the virtual machine.

*A. Network Performance*

In this subsection, we compare the network performances with ND-CRV and CD-CRV schemes and explain the reasons for those diversities. Figs. 10–12 compare the performances of these two schemes in terms of average mistrust rate, resource utilization, and path provisioning latency.

As shown in Fig. 10, the average mistrust rate is evaluated in different traffic loads, which shows that the controllers’ average mistrust rate in CD-CRV scheme is less than that of ND-CRV scheme. With the comparison of the distrust rate of the two CRV schemes, we will elaborate on the differences and the consequent impact between the two algorithmic processes. Here, controllers in ND-CRV scheme first select paths and then provide the optimal path for routing verification. Once a computing error happens on the controller or the path verification fails, it will

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:22:27 UTC from IEEE Xplore. Restrictions apply.

7102 IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 16, NO. 11, NOVEMBER 2020

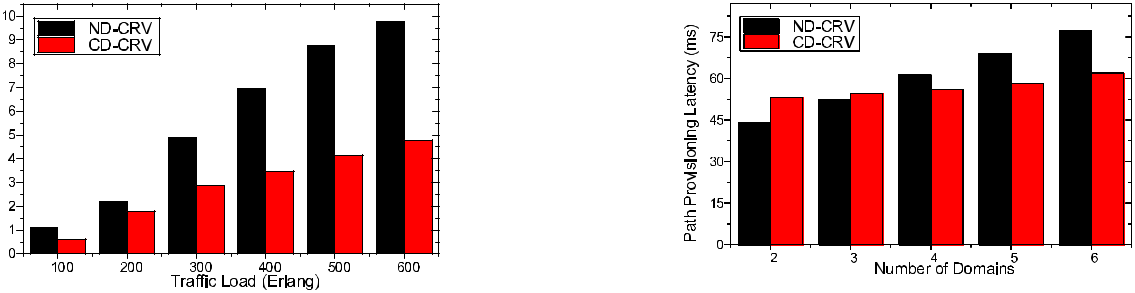
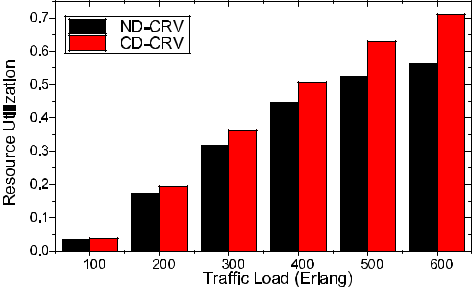


Fig. 12. Provisioning latency between ND- CRV and CD-CRV.

Fig. 10. Average mistrust rate between ND- CRV and CD-CRV.

Fig. 11. Resource utilization between ND- CRV and CD-CRV.



be regarded as mistrust of the controller. On the contrary, in the CD scheme, controllers provide a routing set for routing verification, from which the verifier can obtain a feasible routing in most cases. Thus, in the CD-CRV scheme, the approach of verification first and then path selection provides fault tolerance for routing computing and verification, so the average mistrust rate is lower than ND-CRV scheme.

Fig. 11 shows that the CD-CRV scheme is superior to the ND-CRV scheme in resource utilization, especially when the network is under a heavy traffic load. The reason for this phenomenon is similar to that in the average mistrust rate. Controllers in CD-CRV scheme perform the path selection and global routing combination in the end. At this point, the controller which gains the final verified candidate routing sets can select a more cost-efficient multidomain routing from a global perspective. However, ND-CRV is a scheme that first implements path selection and then verifies the single path, which only considers the load balance in one domain, so it performs worse than CD-CRV in the global resource load.

Fig. 12 shows path provisioning latency between ND-CRV and CD-CRV in MEC collaboration task. In these tasks, the MEC server performs computing independently and returns the results to user devices, and then, performs data interaction with other servers through the provisioning cross-domain routing to complete operations such as big data analysis, data migration, etc. As shown in Fig. 12, ND-CRV scheme can provide a multidomain MEC collaboration routing for the service more quickly when the network scale is small. However, as the number of domains increases, the path provisioning latency of ND-CRV scheme increases faster than that of CD-CRV scheme, and CD-CRV scheme has a lower latency when the MEC network

has four domains. The path provisioning latency of the two CRV algorithms is mainly generated by routing computation and routing verification, which corresponds to the black and red parts in Figs. 4 and 5, respectively. From Figs. 4 and 5, we can see that the routing verification parts of the two CRV schemes are approximately the same, so the latency they generate will not be much different. Therefore, it can be inferred that the path provisioning latency difference between the two CRV schemes mainly occurs in the routing computing. When the network size is small, ND-CRV scheme requires fewer path-computing oper-ations, so the processing delay is low. On the other hand, when the network size is large, CD-CRV executes routing computation in parallel, which saves a lot of signaling and queuing delay, so the path provisioning latency is low.

*B. Verification Performance*

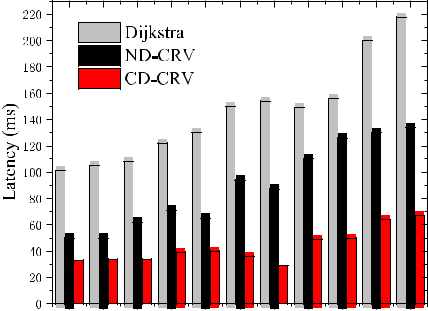
Facing the increasing network scale and data volume, the error detection latency of MEC collaboration must be considered as well. In this subsection, we evaluate BlockTC with ND-CRV and CD-CRV in terms of error detection latency. Besides, the two CRV schemes compute routing based on resource weights, which is different from the Dijkstra algorithm. Dijkstra algo-rithm is the most widely used routing algorithm in the current communication network, which can effectively resolve the short-est path problem in most network topologies, especially for the mesh and chain topologies. In addition, on the premise of the shortest path, Dijkstra algorithm can also achieve more flexible routing computation by adding certain evaluation factors such as the cost of nodes or edges, which is similar to the design characteristics of our scheme. Therefore, we select the Dijkstra algorithm as a comparison in the simulation to represent the common systems.

Fig. 13 demonstrates the error detection latency about ND-CRV and CD-CRV schemes proposed in this article as well as Dijkstra algorithm which is widely implemented in main-stream routing solutions. It can be seen that the CRV schemes significantly outperform Dijkstra algorithm in the error detection latency. This is because the CRV schemes can detect the routing error during the routing verification consensus at the path provi-sioning procedure. However, Dijkstra algorithm can only detect the error routing from current traffic status without revealing the network topology to other domains for verification. In the comparison of ND-CRV and CD-CRV schemes, the CD-CRV scheme has obvious advantages in error detection latency. This

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:22:27 UTC from IEEE Xplore. Restrictions apply.

|  |  |
| --- | --- |
| YANG *et al.*: DISTRIBUTED BLOCKCHAIN-BASED TRUSTED MULTIDOMAIN COLLABORATION FOR MOBILE EDGE COMPUTING | 7103 |
|  |  |

Fig. 13. Error detection latency of BlockTC.



is because CD-CRV uses parallel methods for path computing and verification, which can detect errors faster in distributed consensus. In addition, as the traffic load increases, the serial method used by the ND-CRV scheme will accumulate a large amount of queuing latency. Therefore, the gap between the error detection latency of the two schemes is as small as that of the path provisioning latency under low traffic load, but there will be a significant gap between the error detection latency of the two schemes under high traffic load.

VI. CONCLUSION

In order to meet the demands of high mobility and low latency of new 5G services, MEC has gradually developed into heteroge-neous systems, which poses severe challenges to the credibility and privacy protection of cross-domain collaboration of MEC. This article showed that the combination of blockchain and distributed multidomain networks can effectively accomplish the trusted cross-domain collaboration and topology privacy protection of MEC systems. Specially, we used backup dual links and ABF for distributed ledger maintenance and efficient routing verification. Backup dual links extracted weighted vir-tual topology from physical topology, which provided a basis for cross-domain routing verification in MEC collaboration and prevented physical topology privacy exposure. ABFs were used as the carrier of routing verification. Its special hash retrieval and adaptive performance made MEC collaboration become more efficient and adaptive to the dynamic changes of network topology. Based on the testbed, we simulated the BlockTC and evaluated its performance in mistrust rate, resource utilization, path provisioning latency, and error detection latency, which demonstrated the feasibility and advantage of the proposed architecture.

REFERENCES

1. H. Arasteh *et al.*, “IoT-based smart cities: A survey,” in *Proc. IEEE 16th* *Int. Conf. Environ. Elect. Eng.*, Florence, Italy, 2016, pp. 1–6.
2. “Cisco global cloud index: Forecast and methodology 2015–2020,” Cisco, San Jose, CA, USA, White Paper, 2016.
3. A. Botta *et al.*, “Integration of cloud computing and Internet of Things: A survey,” *Future Gener. Comput. Syst.*, vol. 56, pp. 684–700, 2016.
4. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge comput-ing: Vision and challenges,” *IEEE Internet Things J.*, vol. 3, no. 5,
   1. 637–646, Oct. 2016.
5. M. Satyanarayanan, “The emergence of edge computing,” *Computer*, vol. 50, no. 1, pp. 30–39, 2017.
6. Y. C. Hu *et al.*, “Mobile edge computing—A key technology towards 5G,” ETSI, Sophia Antipolis, France, White Paper No. 11, 2015.
7. L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, “Challenges on wireless heterogeneous networks for mobile cloud computing,” *IEEE Wireless* *Commun.*, vol. 20, no. 3, pp. 34–44, Jun. 2013.
8. T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, “A cooperative scheduling scheme of local cloud and Internet cloud for delay-aware mobile cloud computing,” in *Proc. IEEE Globecom Workshops*, 2015, pp. 1–6.
9. F. Wang, J. Xu, X. Wang, and S. Cui, “Joint offloading and computing optimization in wireless powered mobile-edge computing systems,” *IEEE* *Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.
10. W. Yu *et al.*, “A framework to enable multiple coexisting Internet of things applications,” in *Proc. Int. Conf. Comput., Netw. Commun.*, 2018,
    1. 637–641.
11. R. Yu, J. Ding, S. Maharjan, S. Gjessing, Y. Zhang, and D. H. K. Tsang, “Decentralized and optimal resource cooperation in geo-distributed mobile cloud computing,” *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1,
    1. 72–84, Jan./Mar. 2018.
12. H. Yang *et al.*, “BlockONet: Blockchain-based trusted cloud radio over optical fiber network for 5G fronthaul,” in *Proc. Opt. Fiber Commun.* *Conf. Expo.*, 2018, pp. 1–3.
13. Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, “Consortium blockchain for secure energy trading in industrial Internet of Things,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018.
14. K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, “Privacy-preserving energy trading using consortium blockchain in smart grid,” *IEEE Trans. Ind.* *Informat.*, vol. 15, no. 6, pp. 3548–3558, Jun. 2019.
15. J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A sur-vey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications,” *IEEE Internet Things J.*, vol. 4, no. 5,
    1. 1125–1142, Oct. 2017.
16. X. Yang, T. Wang, X. Ren, and W. Yu, “Survey on improving data utility in differentially private sequential data publishing,” *IEEE Trans. Big Data*, to be published, doi: [10.1109/TBDATA.2017.2715334](https://dx.doi.org/10.1109/TBDATA.2017.2715334).
17. C. Wang, K. Ren, and J. Wang, “Secure optimization computation out-sourcing in cloud computing: A case study of linear programming,” *IEEE* *Trans. Comput.*, vol. 65, no. 1, pp. 216–229, Jan. 2016.
18. Z. Qin *et al.*, “Heavy hitter estimation over set-valued data with local differential privacy,” in *Proc. ACM SIGSAC Conf. Comput. Commun.* *Secur.*, 2016, pp. 192–203.
19. Z. Xiong, S. Feng, D. Niyato, P. Wang, and Z. Han, “Optimal pricing-based edge computing resource management in mobile blockchain,” in *Proc.* *IEEE Int. Conf. Commun.*, 2018, pp. 1–6.
20. H. Yang *et al.*, “Distributed blockchain-based trusted control with multi-controller collaboration for software defined data center optical networks in 5G and beyond,” in *Proc. Opt. Fiber Commun. Conf. Expo.*, 2019,
    1. 1–3.
21. H. Yang *et al.*, “Blockchain-based trusted authentication in cloud radio over fiber network for 5G,” in *Proc. 16th Int. Conf. Opt. Commun. Netw.*, 2017, pp. 1–3.
22. Y. Liang *et al.*, “Blockchain-based efficient recovery for secure distributed control in software defined optical networks,” in *Proc. Opt. Fiber Commun.* *Conf. Expo.*, 2019, pp. 1–3.
23. H. Yang *et al.*, “Blockchain-based hierarchical trust networking for Joint-Cloud,” *IEEE Internet Things J.*, 2019, doi: [10.1109/JIOT.2019.2961187](https://dx.doi.org/10.1109/JIOT.2019.2961187).
24. A. Singh *et al.*, “Bloom filter based optimization scheme for massive data handling in IoT environment,” *Future Gener. Comput. Syst.*, vol. 82,
    1. 440–449, 2018.
25. H. Yang *et al.*, “Blockchain-based secure distributed control for software defined optical networking,” *China Commun.*, vol. 16, no. 6, pp. 42–54, 2019.
26. D. C. Nguyen *et al.*, “Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning,” 2019, *arXiv:1908.07467*.
27. M. A. Rahman *et al.*, “Blockchain-based mobile edge computing framework for secure therapy applications,” *IEEE Access*, vol. 6,
    1. 72469–72478, 2018.
28. H. Yang *et al.*, “Experimental demonstration of multi-dimensional re-sources integration for service provisioning in cloud radio over fiber network,” *Sci. Rep.*, vol. 6, 2016, Art. no. 30678.

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:22:27 UTC from IEEE Xplore. Restrictions apply.

7104 IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 16, NO. 11, NOVEMBER 2020

1. H. Yang, J. Zhang, Y. Ji, and Y. Lee, “C-RoFN: Multi-stratum resources optimization for cloud-based radio over optical fiber networks,” *IEEE* *Commun. Mag.*, vol. 54, no. 8, pp. 118–125, Aug. 2016.
2. H. Yang *et al.*, “Bandwidth compression protection against collapse in fog-based wireless and optical networks,” *IEEE Access*, vol. 6,
   1. 54760–54768, 2018.
3. I. Nosheen, S. A. Khan, and F. Khalique, “A mathematical model for cross layer protocol optimizing performance of software-defined radios in tactical networks,” *IEEE Access*, vol. 7, pp. 20520–20530, 2019.
4. W. Mo *et al.*, “Deep neural network based dynamic resource reallocation of BBU pools in 5G C-RAN ROADM networks,” in *Proc. Opt. Fiber* *Commun. Conf. Expo.*, 2018, pp. 1–3.
5. H. Li, K. Ota, and M. Dong, “Learning IoT in edge: Deep learning for the Internet of Things with edge computing,” *IEEE Netw.*, vol. 32, no. 1,
   1. 96–101, Jan./Feb. 2018.
6. S. Thomas *et al.*, “Joint design of Dijkstra’s shortest path routing and sleep-wake scheduling in wireless sensor networks,” in *Proc. Int. Conf.* *Energy, Commun., Data Analytics Soft Comput.*, 2017, pp. 981–986.
7. S. Broumi, A. Bakal, M. Talea, F. Smarandache, and L. Vladareanu, “Ap-plying Dijkstra algorithm for solving neutrosophic shortest path problem,” in *Proc. Int. Conf. Adv. Mechatronic Syst.*, 2016, pp. 412–416.

**Hui Yang** received the Ph.D. degree from Bei-jing University of Posts and Telecommunica-tions, Beijing, China.



He is currently Vice Dean of Institute of Sci-ence and Technology and an Associate Pro-fessor of State Key Laboratory of Information Photonics and Optical Communications with the Beijing University of Posts and Telecommunica-tions, Beijing, China. He has authored or coau-thored 100 papers in prestigious journals and conferences, and is the first author of more than

50 of them. His research interests include software defined network (SDN), fixed-mobile access networks, data center network, blockchain, and so on.

Dr. Yang was the recipient of the Best Paper Award at the IEEE In-ternational Conference on Wireless Communications and Mobile Com-puting 2019/Symposium on Network Cloud Computing and Applications 2015 and the Young Scientist Award at the IEEE International Confer-ence on Optical Communications and Networks 2017. He was the Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, Asso-ciate Technical Editor (ATE) of IEEE COMMUNICATIONS MAGAZINE, and General Chair of the International Conference on Information System and Artificial Intelligence, 2016.

**Yonshen Liang** was born in Guangdong,China, in 1994. He received the B.S. degree in communication engineering from South China Normal University (SCNU), Guangdong, China, in 2017, and is currently working toward the M.S. degree in electronics and communication engineering with the Beijing University of Posts and Telecommunications, Beijing. China.



His research interests include blockchain, software-defined optical networking, and 5G.

**Jiaqi Yuan** was born in Guangdong, China, in1995. He received the B.S. degree in commu-nication engineering from South China Normal University, Guangdong, in 2018. He is currently working toward the M.S. degree in electronics and communication engineering with the Beijing University of Posts and Telecommunications, Beijing, China.



His research interests include blockchain, software-defined optical networking, and 5G.

**Qiuyan Yao** received the M.S. degree in com-puter science and technology from the Hebei University of Engineering, Handan, China, in 2015. She is currently working toward the Ph.D. degree in communication and information sys-tem with the Beijing University of Posts and Telecommunications, Beijing, China.



Her research interests include the routing and spectrum assignment strategies in elastic opti-cal networks and software-defined optical net-works.

Ms. Yao is a Member of The International Society for Optical Engi-neering (SPIE).

**Ao Yu** was born in Shandong, China, in 1992.He received the B.S. degree in communica-tion engineering from the China University of Petroleum, Shandong, in 2014. He is currently working toward the Ph.D. degree in information and communication engineering with the Beijing University of Posts and Telecommunications, Beijing, China.



His main research interests include resource allocation, data center optical networks, and deep learning.

**Jie Zhang** received the Ph.D. degree from Bei-jing University of Posts and Telecommunica-tions, Beijing, China.



He is currently a Professor of State Key Lab-oratory of Information Photonics and Optical Communications and Dean of the Institute of Information Photonics and Optical Communica-tions, Beijing University of Posts and Telecom-munications (BUPT), Beijing, China. He is spon-sored by more than ten projects of the Chinese government. He has authored or coauthored

eight books and more than 100 articles, and seven patents have also been granted. His research interests include optical transport networks, packet transport networks, and so on.

Prof. Zhang was a Technical Program Committee Member for Asia Communications and Photonics Conference (ACP) 2009, International Conference on Photonics in Switching (PS) 2009, Conference on Optical Network Design and Modeling (ONDM) 2010, and so on.

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:22:27 UTC from IEEE Xplore. Restrictions apply.