IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Prediction Algorithm For Fixed Identity Mapping In An Edge Computing Environment

Nakul Ashok Gade
MVPS's Rajarshi Shahu Maharaj Polytechnic, Nashik

Vikas Pralhad Gawai
MVPS's Rajarshi Shahu Maharaj Polytechnic, Nashik

Ankita Kamlakar Pangavhane
MVPS's Rajarshi Shahu Maharaj Polytechnic, Nashik

Sachin Ashok Surywanshi
MVPS's Rajarshi Shahu Maharaj Polytechnic, Nashik

ABSTRACT

A research hotspot that extends cloud computing to the network's edge is edge computing. The ability to integrate the locator/identity separation protocol (LISP) into edge networks has been made viable by recent advancements in end device computation, storage, and network technology. As a result, we incorporate LISP into edge routers at the edge network in this study, concentrating especially on the delay problem of mapping resolution and cache updating in LISP with the aid of edge computing. We initially examine how the locator/identity separation network communicates before considering applying the prediction approach to support this research in order to address the delay problem. We propose and develop a Fixed Identity Mapping Prediction Algorithm (FIMPA) based on collaborative filtering in order to get good prediction results, and we further confirm the efficacy of the proposed algorithm by experiments on real-world data.

Keywords

FIMPA, NLP, machine translation, machine learning, computational techniques, linguists

INTRODUCTION

Recent years have seen the emergence and widespread use of a growing number of computing technology innovations (such as cloud computing [1], [2], cluster computing, IPTV [3], etc.). While the terminal is generally thin and occasionally even has no functions to process, they are always built so that the majority of functions are executed in the core (datacenter). The strategy used by edge computing is to offload some processes or functions to terminals from the core due to advancements in the communication and storage capabilities of terminal devices [4]–[6]. Pushing computation, information, memory, and networking away from centralised nodes to the logical edges of a network constitutes edge computing, a growing area of research and an extension of cloud computing. By processing data at the network's edge, close to the data source, edge computing is an effective way to maximize cloud computing.

By performing computation and storage at or close to the original location of the data, it (a) reduces the communication bandwidth needed between edge nodes and the data centre; (b) it may limit or remove a significant bottleneck as well as a potential failure point in the cloud computing environment; and (c) it improves data security because data is encrypted before being moved to the core network because the majority of edge nodes are virtualized, it achieves strong scalability. Edge computing-related fog computing was first proposed by Cisco [7, 8]. In order to facilitate the operation of compute, memory, and networking between end devices and cloud storage systems, cloud computing is extended to the edge of an enterprise's network.

In order to address concerns with mobility, multi-homing, and the IP semantic overload problem, Cisco researchers devised and implemented the LISP protocol in certain of their routes deployed in networks. The LISP protocol divides already-existing IP addresses into entity identities (EID) and router identifier/locators (RLOC); additionally, it explains the locator/identity separation protocol from a network perspective, which means that it can stop the end hosts' network protocol 17356. The Open Source Attribution 4.0 License governs the use of this work. For further details, see VOLUME 8, 2020 architecture from altering while just changing the operating mode of network devices (http://creativecommons.org/licenses/by/4.0/). EID is used to identify an end host in a LISP network while connecting with other hosts. It is not reliant on the network structure.

Additionally, since the edge network now has access to the datacenter's computation, memory, and networking resources, it is also possible for the edge network to address the long-standing issues with LISP, such as the delay in mapping resolution, the storage space limitations for mapping entries, and so forth. Thus, it represents a workable method of fusing edge computing and LISP. Our goal is to find solutions to some integration process problems.

With the help of edge computing, we focus on the delay problem of mapping resolution; this problem arises when an identity cannot be resolved locally, which triggers a mapping resolution request to the mapping system and waits for the response.

The communication process of the Locator/Identity separation network is initially examined in this study, with particular attention paid to the update mechanism for mapping the cache in edge routers and the latency problem of mapping resolution in LISP. Then, using collaborative filtering, we suggest the Fixed Identity Mapping Prediction Algorithm (FIMPA). Lastly, we conduct experiments to confirm the performance of our suggested algorithm. The results of the experiment show how the FIMPA algorithm may considerably increase hit rate and decrease delay.

1. Background

A. EDGE COMPUTING

Cloud computing has been the subject of a lot of research in recent years. Cloud computing is an on-demand computing architecture that offers on-demand access to a shared pool of configurable networked resources (such as CPUs, storage, VMs, networks, apps, and servers) that can be deployed quickly and with little administration labour. On a subscription basis, service providers make clouds with predetermined quality of service (Q'oS) terms available to interested clients through the Internet, giving them access to a variety of simple-to-use, affordable services. Cloud computing is a server-centric computing paradigm in which practically all operations are conducted in the core (datacenter), while the client is very thin and may even have no processing capacity.

Although cloud computing has many benefits, such as simple maintenance, centralised management, and high server utilization, its drawbacks have been revealed in the era of mobile Internet, such as the need for terminals with higher processing power and security issues, among other things. In light of this, transparent computing [16], [17] puts out a possible answer for the mobile Internet. The fundamental tenet is that all information, including operating systems, applications, and user data, is stored on servers and processed by terminals. This strategy has several benefits, including lower terminal complexity and expense, better user experience, high-level security, and cross-platform application compatibility [17]. Access control can use this as well [18].

Due to the Internet of Things' (IoT) rapid development over the past three years, edge computing has emerged as a research hotspot. This is because applications, data, and networking services are moving from centralised nodes in datacenters to end devices. By offloading some tasks or functions to clients from the core, this approach takes advantage of the powerful computation and storage capabilities of the terminal. Doing so has a number of benefits, including lowering the communications bandwidth between edge nodes and the datacenter, removing bottlenecks and potential failure points in the cloud environment, enhancing data security, and achieving good scalability. Further edge computing research, which takes into account using block chain [22], may be found in [19]-[21].

B. CACHE UPDATE MECHANISM

Cache replacement algorithms and cache prediction/prefetching algorithms are the two broad categories into which the related domestic and international research can be separated with regard to the updating mechanism of the mapping cache in edge routers.

The main goal of cache replacement algorithms is to replace the mapping entries in the cache when there is not enough room for them. Either temporal locality or spatial locality is often the focus of research in this field.

A conventional technique called LRU (Least Recently Used) [23] makes the assumption that recently visited objects are most likely to be revisited in the future. As a result, it always takes the oldest object in the cache that has not been accessed. Although LRU is the most often used algorithm of its kind, it is also the simplest; nonetheless, because it takes into account the object time factor, its efficacy is not very high. The LRU-2 algorithm is improved by the 2Q (Two queues) algorithm [25], which separates cached pages into "warm" and "cold" queues before caching them in two FIFO queues. A page will be added to the cold page queue when it is first accessed; if the viewable page is already in the cold queue, nothing will happen.

Additionally, the LFU (Least Frequently Used) algorithm [26] takes the most recent least-visited objects out of the cache by taking full advantage of the previous scheduling data in the cache and considering the access frequency of recent objects. The LFU algorithm fully exploits the characteristics of user access frequency to favoured resources, but it is unable to differentiate between objects that are frequently accessed in the early or later stages. Moreover, it could keep "expired" objects in the cache to take up space, creating a significant cache pollution issue. The LFU- Aging method is an enhanced version of LFU's optimization algorithm that addresses one of its main issues, namely the fact that resources in the cache are continually being used more frequently and never less frequently. LFU-Aging [27] suggests that the value of access frequency is inversely related to the survival time by taking into account both the access frequency and the survival period of the resource in the cache. The cache's long-lost resources will eventually see fewer and fewer accesses until they are completely gone. Although the two techniques are predicated on geographical locality, there are specific situations where they fail.

Cache prediction/prefetching algorithms can predict which resources will be accessed in the future using the current access, push the prediction content to the local cache, and replace the mapping item in the cache using the cache replacement algorithm. This is done based on the spatial locality of the resources. The prediction hit rate of the prefetching method is closely correlated with that of the prediction model since it uses a prediction model to characterise the mapping request. Prefetching models based on data mining [28], multitask [29], web semantics [30], the Markov model [24], probabilistic model [31], and other techniques are currently widely utilised. The prefetching model based on Web semantics extracts the feature key and analyses user behaviour to predict the next request, whereas the prefetching model based on data mining predicts the users' potential next page by mining a large amount of potential information contained in the users' browsing history. Additionally, the Markov model-based approach employs the transition probability matrix to characterise users' request behaviour and forecasts users' subsequent requests. The precision of the prediction algorithm, however, determines how good these algorithms are. The technique described in [32] can be used to retrieve the feature key.

2. Network Issues

We first examine the Locator/Identity Separation network's communication flow in order to address the delay issue. The delay issue with LISP is then covered.

A. COMMUNICATION PROCESS OF THE LISP NETWORK

Like in regular networks, there are two categories of end hosts in a LISP network: fixed hosts and mobile hosts. These two host types must be easily distinguished from one another.

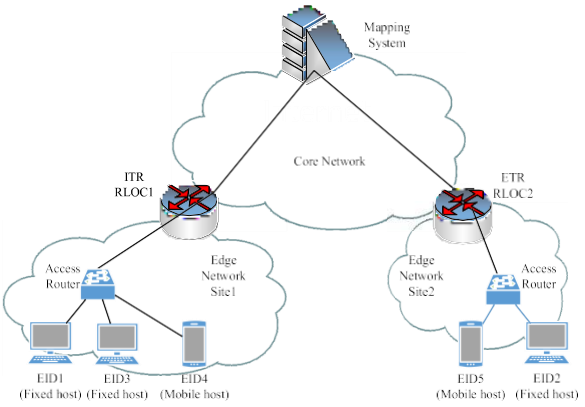

Fig 1. An instance of Location/Identity separation network.

The LISP protocol uses a specific EID segment to represent mobile nodes, while the remaining address segments are assigned to fixed nodes. Considering the fixed hosts managed by an access router, their EIDs can be allocated regularly and aggregated into one EID-prefix, which can be used for EID mapping resolution. The mapping system contains the EID-to-RLOC or EID-prefix-to-RLOC mapping entries for mobile hosts and stationary hosts, respectively. The RLOC, which is typically associated with the network topology, is the edge router's IP address used for packet forwarding in the core network. A router's local cache or mapping system, as depicted in Fig. 1, can be used to query the EID-to-RLOC mapping database to determine the RLOC. The Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR) concepts are also introduced by LISP. These devices are used to encapsulate and DE encapsulate packets for identifier communication. ITR is the sender end host's first-hop access router, whereas ETR is the receiver end host's last-hop access router. The EID of both communication parties is contained in the source and destination addresses of a LISP packet that is received by ITR from one end host. To find the RLOC of the access router used to serve the destination end host, ITR uses the destination EID as the keyword to query the EID-to-RLOC mapping, either locally or remotely. Then, ITR creates a new packet and sends it to the core network while including this LISP packet in an outer header.

The source address for the outer header is the RLOC of the ITR, and the destination address is the RLOC of the access router at the destination. The inner LISP packet is forwarded to the destination end host provided by the destination EID by ETR when it gets this new packet that ITR has enclosed. The ITR local buffer will wait a long time to receive the mapping response from the mapping system after initiating a mapping query request to the mapping system if there is no mapping entry EID-to-RLOC of the destination EID. This increases the delay in the mapping resolution and affects the communication quality and performance. The answer to the mapping resolution delay problem is the main problem at hand. A data-driven approach to modelling the Internet route has been suggested [10], evolutionary game theory has been applied to the Internet of Vehicles [11], and transformation-based processes have been applied to IoT [12]. Future applications of our approach include a smart campus [13] and privacy concerns [14].

B. THE DELAY PROBLEM IN LISP

We primarily employ the local mapping cache and identity mapping prediction/prefetching technology in the study of LISP architecture to address the identity mapping resolution delay issue. The edge router must save the identity received from mapping resolution's mapping entry in its local cache in order for the local mapping caching technique to work. This technique can decrease sending the mapping resolution request to the mapping system in order to reduce the mapping resolution latency because it exploits the temporal locality principle of map- ping queries. The cache performance introduced by the upgrade cannot, however, satisfy the performance requirements due to the growing frequency of updates to network resources. Therefore, mapping prediction/prefetching technology is provided to further reduce the mapping resolution latency. The source address for the outer header is the RLOC of the ITR, and the destination address is the RLOC of the access router at the destination. The inner LISP packet is forwarded to the destination end host provided by the destination EID by ETR when it gets this new packet that ITR has enclosed. The ITR local buffer will wait a long time to receive the mapping response from the mapping system after initiating a mapping query request to the mapping system if there is no mapping entry EID-to-RLOC of the destination EID. This increases the delay in the mapping resolution and affects the communication quality and performance. If EID2 has no local mapping entry, the ITR will make a mapping resolution request to the mapping system and, upon receiving the mapping response, store a new mapping entry in its local buffer. In order to create a new packet (packet2), the ITR encapsulates packet1 with an outer packet header. Packet2 selects RLOC1 (the RLOC of the ITR) as the source address and RLOC2 (the RLOC of the ETR) as the destination address. The ITR then sends packet 2 to the core network. According to the destination address RLOC2, this new packet (packet2) is forwarded across the core network to the ETR of Site 2 According to the destination identification EID2, the ETR decapsulates this new packet (packet2) and delivers the LISP packet (packet1) to Host EID2. The ITR will start a mapping query request to the mapping system if no mapping entry EID-to-RLOC of EID2 is detected in the local mapping buffer. It may then have to wait a long time to receive the mapping response from the mapping system. As a result, this could lengthen the mapping resolution delay and impact the effectiveness and quality of communication. The mapping resolution delay might be significantly decreased if we could properly forecast the destination identifiers that a user will connect with in the future and the mapping system permitted pushing the mapping entries of specific identifiers to edge routers. To further reduce this latency, we may further combine the technologies of caching and forecasting. Someone connected to a college network, for instance, would routinely use Google Scholar or Badu Academic to look for scholarly publications. Similarly, if the access resources used by people in two edge networks are comparable, it makes sense to direct people in the second edge network to use the resources that people in the first network commonly use. Future work will focus on developing an efficient mapping management system employing key management techniques found in sensor networks [34, 35], block chain technology [36], and deep learning [37] to address unsolved problems.

.

3. The Fixed Identity Mapping Prediction Algorithm

The edge router TR (denoting either ITR or ETR in an edge network) in a locator/identity separation network receives a packet from its inner interface (that is connected to devices in the edge network). For the destination identifier of the packet, it will first check the local cache for the relevant EID-to-RLOC (for mobile EID) or EID-prefix-to-RLOC (for fixed EID) mapping item. A mapping resolution request will be sent to the mapping system if the local cache is unable to locate this linked mapping entry. As all of the network's import and export flows transit via edge routers, they are able to keep track of every resource that users in this edge network access. Recent recommendations for methodologies and algorithms include cooperative approaches based on collaborative filtering . The core idea behind these methods is to use group wisdom for prediction and recommendation, deter- mine the relevance of users or items by using information, e.g. users' hobbies, and then making predictions and recommendations based on the correlation. Collaborative filtering is divided into user-based collaborative filtering [9], item- based collaborative filtering and matrix decomposition-based collaborative filtering. In a locator/identity separation network, the users in an edge network have access preferences regarding network resources, causing some edge networks to have a correlation. We can use the idea of collaborative filtering to predict which resources in an edge network will be accessed next time and thus push the corresponding identity mapping to edge routers in advance. These approaches' central tenet is the use of collective wisdom for prediction and suggestion. Relevance of individuals or products is

determined using information, such as users' hobbies, and predictions and recommendations are then made based on the correlation. The three types of collaborative filtering are matrix decomposition-based collaborative filtering, item-based collaborative filtering, and user-based collaborative filtering [9]. Users in an edge network in a locator/identity separation network have access preferences for network resources, which leads to a correlation in some edge networks. We may push the necessary identity mapping to edge routers in advance by using the concept of collaborative filtering to forecast which resources in an edge network will be accessed the next time. For the fixed end hosts situation, we apply the collaborative filtering notion and refer to this as the Fixed Identity Mapping Prediction Algorithm (FIMPA). The FIMPA concept may be summed up as follows:

- Collect the packets received by an edge router over a specified time period; obtain the request frequency of various EID-prefixes over this time period using packet statistics;
- Upload the request frequency information to the mapping system;
- The mapping system then calculates the prediction model based on the collaborative filtering method and generates a fixed identification prediction for the edge router based on the front prediction model;
- The mapping system then collects the packets received by an edge router over a specified time period.

A. SOME CRITICAL CONSIDERATIONS IN THE FIMPA ALGORITHM

1. THE FREQUENCY STATISTICS OF THE FIXED EID-PREFIX FROM THE HISTORY PACKETS

We may identify the data correlation, which serves as the foundation for forecasting and prediction, by analyzing the request frequency statistics of the EID-prefixes recorded in an edge router during a certain time period. So, it's crucial to compile all of the packets that an edge router has received throughout the predetermined time frame. In the case of fixed hosts, the edge router maps the destination identification of each packet to the relevant EID-prefix and checks to see if a local cache entry exists that maps the EID-prefix to the RLOC. So, throughout the defined time period, we may log the corresponding request frequency for the EID-prefixes that the edge router accessed. The EID-prefix request frequency data is subsequently forwarded by the edge router to the mapping system for centralized processing. In order to access various EID-prefixes, the mapping system constructs an array based on the information provided by all edge routers in the edge networks on the frequency of EID-prefix request. This range is shown in Table 1. In this array, "S1, S2,..., Sm" stands for the set of m LISP network edge routers, whereas "D1, D2,..., Dn" stands for the collection of n EID-prefixes. A matrix, mn, which indicates that the edge router Si has accessed the EID-prefix Dj Hij times, represents the request frequency statistics of an EID-prefix accessed by an edge router.

2. MAKING PREDICTIONS BASED ON THE REQUEST FREQUENCY INFORMATION

The mapping system may perform collaborative filtering-based EID-prefix prediction for the fixed end hosts after gathering the EID-prefix frequency data from all edge routers in the LISP network. More particularly, we handle all EID-prefix request frequency information using matrix decomposition-based collaborative filtering. By counting the number of times an edge router visits a certain EID-prefix, one might infer the relationship between the two. In other words, the chance that the edge router will visit this EID-prefix again increases as the number of times it accesses this EID-prefix increases. As an edge router often cannot access the entire collection of EID-prefixes, it can be seen that the request frequency of all EID-prefixes in one edge router is not all non-zero, meaning that the EID-prefix request frequency matrix Rm∗n is sparse.

B. THE INTRODUCTION OF THE FIMPA ALGORITHM

An overview of the algorithm is presented in Algorithm 1. Some details of the implementation are as follows:
Step 1: randomly generate a U(0), which can be set as the global mean.
Step 2: fix U(0), and solve V(0).
At this point, the loss function can be expressed via the following equation:
i 3 4 u1, u2, . . . , um are calculated in accordance with (11); thus, we obtain U(1), which consists of u1, u2, , um.
The FIMPA algorithm loops the execution of Step 2 and 3, and stops after iterating N times. Following the execution, we obtain the optimal solution U and V. The sum of the optimal solution U and V complements the missing value for the request frequency matrix R. For the edge router x, we select the corresponding row of x from the matrix UVT and sort the elements in the row from large to small after removing the original values. The larger the value, the greater the likelihood that edge router x will make an access request to this EID-prefix in the future. We then take the Top-N EID-prefixes as the prediction for

$$C = \Sigma [(rij - (u(0))vj)2 + \lambda(||u(0)||2 + ||vj||2)] \quad (2$$

Fixing j (j = 1,2, . . . ,n), the derivative of C is: $\partial C = 2 \Sigma [((u(0))T u(0)) + \lambda)vj - riju(0)]$ (3)table timer. When the setting time is reached, the map- ping system generates EID-prefix-to-RLOC mapping entries according to the FIMPA algorithm and actively pushes them ∂vj. to the edge routers. When the edge routers receive these EID- prefix-to-RLOC mapping entries, they will update the map- Let $\partial C = 0$; thus, we get $\Sigma [((u(0))T u(0) + \lambda)vj] = \Sigma riju(0)$(4).ping cache according to their cache replacement strategy (e.g. LRU and LFU). It can be improved using semi-supervised learning [38] in the future. The FIMPA algorithm decomposes the request frequency matrix into the product of two small matrices: the character- Equation (4) is equivalent to (UUT + λE)vj = UrT (5) Matrix of an edge router and the characteristic matrix of EID-prefixes. It then carries out the prediction based on the Algorithm 1 FIMPA Algorithm Input: request frequency matrix R, iteration number T, number of features k, an edge router x Output: the prediction result set S for the edge router

x Data: U stands for the characteristic matrix of an edge router, V for the characteristic matrix of EID-prefixes, map for the key-value set of (EID-prefix i, prediction degree p) The pseudo code of the FIMPA algorithm is as follows: complete value of the product of these two matrices. Algorithm 1 presents the pseudo code of the FIMPA algorithm; here, initM(k) initializes the feature matrix according to the number of features, calV(U) calculates U using V, calU(V) calculates V using U, getUnvisited(R, u) returns a collection of EID-prefixes that edge router u has not accessed, and getResult(map) obtains the prediction by sorting the results. The correction of FIMPA algorithm can be guaranteed by the method of the matrix decomposition-based collaborative filtering, which is proved to be a feasible solution for the recommended problem.

4. Experiments and Evaluation

In this paper, we implement the FIMPA algorithm with LRU and LFU as mapping replacement algorithms, then conduct comparisons with the existing LRU and LFU algorithms In this section, we evaluate the FIMPA algorithm using two indicators: namely, the cache hit rate and the hit rate convergence time. The cache hit rate is the ratio of the number of requests hits in the local cache to the total number of access requests in the time period. The experimental data used in this simulation is real net- work traffic data collected from the Internet, namely the NLANR Auckland-VIII dataset, which represents the identity mapping request traffic. The data format of the dataset is ERF. In total, the dataset contains more than 6 million packets collected over 60 minutes. In order to evaluate the cache performance and use this dataset with the FIMPA algorithm, we need to map the source and destination IP addresses of the packets to the corresponding prefixes. We use the BGP prefix as the EID-prefix and download the BGP core routing table from Route Views, which maps the IP addresses to prefixes. In this paper, we used Java to implement the FIMPA algorithm and ran the simulation program on a desktop PC,

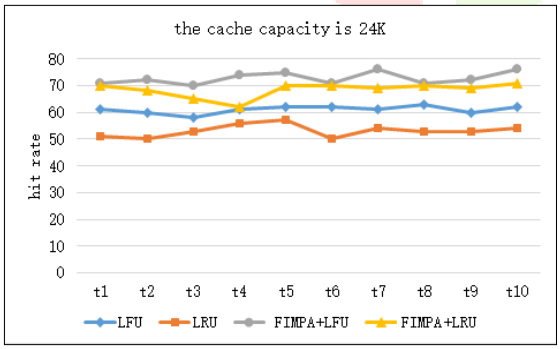

Fig 2. Hit rate results with 24K cache.

Which has an Intel i5-7200U CPU, 8GB memory and the Windows 7 operating system. According to the flow analysis statistics, the simulation sets the statistical period to 11 minutes, and further selects 60 seconds as the prediction time period (hereafter 'time period' for short). The FIMPA algorithm makes a prediction based on historical packets every time period. The packet in the first time period is the initial input data. The packets in the same time period are divided into different subsets according to the address prefix mapped by the source IP address. The different subsets represent the messages received by different edge routers. Subsequently, the packets in each subset are processed as follows: map the destination IP address to the address prefix; count the number of packets belonging to different prefixes; and simulate the frequency with which the edge router accesses the end host. The simulator pushes all predicted results to the mapping cache of the edge router. In the initial case, the mapping cache table is empty and all predicted mappings are saved. For the second time period, the destination IP address of each packet is mapped to the BGP prefix. If the prefix is already in the mapping cache table, the simulator will increase the number of hits by one and process the next message; if no mapping is found in the mapping cache table, the simulator will record the mapping if the table is not full, or alternatively perform a mapping update using a mapping replacement algorithm if the table is full. After processing the second time period, the simulator continues to forecast and push the pro- gram forward using the second time period as the historical data. The subsequent 9 time periods of packet processing are similar.

5. Conclusion and Future Work

By combining the FIMPA prediction algorithm with the replacement strategies (LRU and LFU), the hit rate of the cache mapping table can be significantly improved, especially in the initial state. When the cache mapping table is empty, the hit rate rapidly achieves higher stability in a short convergence time. However, FIMPA gains this improvement in hit rate by sacrificing the algorithmic performance, meaning that the time complexity and spatial complexity of FIMPA are relatively high and increase exponentially with the number of edge routers. To address some security issues in our work, we will consider using technologies such as block chain, the Tor network, covert communications, sensor networks and SDN net- works in our future work. This paper aimed to solve the delay problem for the fixed end hosts; we will consider solutions for the mobile identity case in the future work. Possible solutions may include using a tree storage structure for mobile identity mappings or incorporating block chain technology, SVM algorithm, among others.

6. References

1) Edge Computing. Accessed: 2019. [Online]. Available: https://en. wikipedia.org/wiki/Edge_computing

2) W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, ''Edge computing: Vision and challenges,'' IEEE Internet Things J., vol. 3, no. 5, pp. 637–646, Oct. 2013

3) Y. Yin, L. Chen, Y. Xu, and J. Wan, ''QoS prediction for service recom- mendation with deep feature learning in edge computing environment,'' Mobile Netw. Appl., to be published, doi: 10.1007/s11036-019-01241-7.

4) Fog Computing. Accessed: 2019. [Online]. Available: https://en. wikipedia.org/wiki/Fog_computing

5) I. Stojmenovic, S. Wen, X. Huang, and H. Luan, ''An overview of fog computing and its security issues,'' Concurrency Comput., Pract. Exper., vol. 28, no. 10, pp. 2991–3005, Jul. 2016.

6) Z. Tian, S. Su, W. Shi, X. Du, M. Guizani, and X. Yu, ''A data-driven method for future Internet route decision modeling,'' Future Gener. Com- put. Syst., vol. 95, pp. 212–220, Jun. 2019.

7) Z. Tian, X. Gao, S. Su, J. Qiu, X. Du, and M. Guizani, ‘‘Evaluating repu- tation management schemes of Internet of vehicles based on evolutionary game theory,’’ IEEE Trans. Veh. Technol., vol. 68, no. 6, pp. 5971–5980, Jun. 2019.

8) K. Yan, C. Zhong, Z. Ji, and J. Huang, ‘‘Semi-supervised learning for early detection and diagnosis of various air handling unit faults,’’ Energy Buildings, vol. 181, pp. 75–83, Dec. 2018.

9) H. Gao, Y. Duan, L. Shao, and X. Sun, ‘‘Transformation-based processing of typed resources for multimedia sources in the IoT environment,’’ Wire- less Netw., early access, Nov. 2019, doi: 10.1007/s11276-019-02200-6.

10) Z. Tian, Y. Cui, L. An, S. Su, X. Yin, L. Yin, and X. Cui, ‘‘A real-time correlation of host-level events in cyber range service for smart campus,’’ IEEE Access, vol. 6, pp. 35355–35364, 2018.

11) J. Yu, Z. Kuang, B. Zhang, W. Zhang, D. Lin, and J. Fan, ‘‘Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing,’’ IEEE Trans. Inf. Forensics Security, vol. 13, no. 5, pp. 1317–1332, May 2018.

12) Y. Zhang, K. Guo, J. Ren, Y. Zhou, J. Wang, and J. Chen, ‘‘Transparent computing: A promising network computing paradigm,’’ Comput. Sci. Eng., vol. 19, no. 1, pp. 7–20, Jan. 2017.

13) Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, and Y. Zhang, ‘‘An effi- cient privacy-enhanced attribute-based access control mechanism,’’ Con- currency Comput., Pract. Exper., to be published, doi: 10.1002/cpe.5556.

14) Z. Tian, W. Shi, Y. Wang, C. Zhu, X. Du, S. Su, Y. Sun, and N. Guizani, ‘‘Real-time lateral movement detection based on evidence reasoning net- work for edge computing environment,’’ IEEE Trans. Ind. Inf., vol. 15, no. 7, pp. 4285–4294, Jul. 2019.

15) Y. Xu, G. Wang, J. Ren, and Y. Zhang, ‘‘An adaptive and configurable protection framework against android privilege escalation threats,’’ Future Gener. Comput. Syst., vol. 92, pp. 210–224, Mar. 2019.

16) Y. Xu, G. Wang, J. Yang, J. Ren, Y. Zhang, and C. Zhang, ‘‘Towards secure network computing services for lightweight clients using blockchain,’’ Wireless Commun. Mobile Comput., vol. 2018, Nov. 2018, Art. no. 2051693.

17) K. Yan, W. Shen, Q. Jin, and H. Lu, ‘‘Emerging privacy issues and solutions in cyber-enabled sharing services: From multiple perspectives,’’ IEEE Access, vol. 7, pp. 26031–26059, 2019.

18) Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, ‘‘Blockchain empowered arbitrable data auditing scheme for network storage as a service,’’ IEEE Trans. Serv. Comput., to be published, doi: 10.1109/tsc.2019.2953033.

19) I. Zukerman, D. W. Albrecht, and A. E. Nicholson, ‘‘Predicting users’ requests on the WWW,’’ in Proc. Int. Conf. User Modeling, New York, NY, USA, 1999, pp. 275–284.

20) Johnson, Theodore, Shasha, ‘‘2Q: A low overhead high performance buffer management replacement algorithm,’’ in Proc. VLDB, Santiago, Chile, 1994, pp. 439–450.

21) Y. Yin, J. Xia, Y. Li, Y. Xu, W. Xu, and L. Yu, ‘‘Group-wise itinerary planning in temporary mobile social network,’’ IEEE Access, vol. 7, pp. 83682–83693, 2019.

22) G. Karakostas, O. St, and D. N. Serpanos, ‘‘Practical LFU implementation for Web caching,’’ Princeton Univ., Princeton, NJ, USA, Tech. Rep. TR- 622-00, Jun. 2000.

23) B. Feng, H. Zhou, and G. Li, ‘‘Least popularly used: A cache replacement policy for information-centric networking,’’ J. Internet Technol., vol. 17, no. 1, pp. 1–10, 2016.

24) J. Yu, C. Hong, Y. Rui, and D. Tao, ‘‘Multitask autoencoder model for recovering human poses,’’ IEEE Trans. Ind. Electron., vol. 65, no. 6, pp. 5060–5068, Jun. 2018.

25) H. Gao, W. Huang, and X. Yang, ‘‘Applying probabilistic model checking to path planning in an intelligent transportation system using mobility trajectories and their statistical data,’’ in Proc. AUTOSOFT, Jun. 2019, pp. 547–559.

26) J. Yu, M. Tan, H. Zhang, D. Tao, and Y. Rui, ‘‘Hierarchical deep click fea- ture prediction for fine-grained image recognition,’’ IEEE Trans. Pattern Anal. Mach. Intell., to be published, doi: 10.1109/tpami.2019.2932058.

27) M.-L. Messai and H. Seba, ‘‘A survey of key management schemes in multi-phase wireless sensor networks,’’ Comput. Netw., vol. 105, pp. 60–74, Aug. 2016.

28) A. A. Omar, M. Z. A. Bhuiyan, A. Basu, S. Kiyomoto, andM. S. Rahman, ‘‘Privacy-friendly platform for healthcare data in cloud based on blockchain environment,’’ Future Gener. Comput. Syst., vol. 95, pp. 511–521, Jun. 2019.

29) X. Chen, C. Li, D. Wang, S. Wen, J. Zhang, S. Nepal, Y. Xiang, andK. Ren, ‘‘Android HIV: A study of repackaging malware for evading machine-learning detection,’’ IEEE Trans. Inf. Forensics Security, vol. 15, pp. 987–1001, 2020.

30) X. Yan, B. Cui, Y. Xu, P. Shi, and Z. Wang, ‘‘A method of infor- mation protection for collaborative deep learning under GAN model attack,’’ IEEE/ACM Trans. Comput. Biol. Bioinf., to be published, doi: 10.1109/tcbb.2019.2940583.

31) H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, ‘‘Context- aware QoS prediction with neural collaborative filtering for Internet- of-Things services,’’

IEEE Internet Things J., to be published, doi: 10.1109/jiot.2019.2956827.

32) K. Yan, Z. Ji, H. Lu, J. Huang, W. Shen, and Y. Xue, ''Fast and accurate classification of time series data using extended ELM: Application in fault diagnosis of air handling units,'' IEEE Trans. Syst. Man Cybern, Syst., vol. 49, no. 7, pp. 1349–1356, Jul. 2019.

33) Z. Tian, M. Li, M. Qiu, Y. Sun, and S. Su, ''Block-DEF: A secure digital evidence framework using blockchain,'' Inf. Sci., vol. 491, pp. 151–165, Jul. 2019.

34) Y. Xu, J. Ren, G. Wang, C. Zhang, J. Yang, and Y. Zhang, ''A blockchain- based nonrepudiation network computing service scheme for industrial IoT,'' IEEE Trans. Ind. Inf., vol. 15, no. 6, pp. 3632–3641, Jun. 2019.

35) Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, and Z. Tian, ''Toward a comprehensive insight into the eclipse attacks of Tor hidden services,'' IEEE Internet Things J., vol. 6, no. 2, pp. 1584–1593, Apr. 2019.

36) S. Yan, Y. Cong, S. V. Hanly, and X. Zhou, ''Gaussian signalling for covert communications,'' IEEE Trans. Wireless Commun., vol. 18, no. 7, pp. 3542–3553, Jul. 2019.

37) L. Zhu, X. Tang, M. Shen, X. Du, and M. Guizani, ''Privacy-preserving DDoS attack detection using cross-domain traffic in software defined networks,'' IEEE J. Select. Areas Commun., vol. 36, no. 3, pp. 628–643, Mar. 2018.

38) G.-J. Ra, D. See, M. Z. A. Bhuiyan, and I.-Y. Lee, ''A study on anonymous protocol in a permission blockchain with ensure privacy for a member,'' in Proc. 12th Int. Conf. Secur., Privacy Anonymity Comput., Commun. Storage, Atlanta, CA, USA, Jul. 2019, pp. 456–464.