*Journal of Cyber Security*
DOI:10.32604/jcs.2020.010764
***Article***

Tech Science Press

# Efficient Virtual Resource Allocation in Mobile Edge Networks Based on Machine Learning

**Li Li[1,*], Yifei Wei[1], Lianping Zhang[2] and Xiaojun Wang[3]**

[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Alibaba Cloud Computing, Beijing, China
[3]Dublin City University, Dublin, Ireland
[*]Corresponding Author: Li Li. Email: sigurlily bupt@edu.cn

Received: 26 March 2020; Accepted: 07 September 2020

**Abstract:** The rapid growth of Internet content, applications and services require more computing and storage capacity and higher bandwidth. Traditionally, internet services are provided from the cloud (i.e., from far away) and consumed on increasingly smart devices. Edge computing and caching provides these services from nearby smart devices. Blending both approaches should combine the power of cloud services and the responsiveness of edge networks. This paper investigates how to intelligently use the caching and computing capabilities of edge nodes/cloudlets through the use of artificial intelligence-based policies. We first analyze the scenarios of mobile edge networks with edge computing and caching abilities, then design a paradigm of virtualized edge network which includes an efficient way of isolating traffic flow in physical network layer. We develop the caching and communicating resource virtualization in virtual layer, and formulate the dynamic resource allocation problem into a reinforcement learning model, with the proposed self-adaptive and self-learning management, more flexible, better performance and more secure network services with lower cost will be obtained. Simulation results and analyzes show that addressing cached contents in proper edge nodes through a trained model is more efficient than requiring them from the cloud.

**Keywords:** Artificial Intelligence; reinforcement learning; edge computing; edge caching; energy saving; resource allocation

## 1 Introduction

Mobile edge computing and caching have been proposed to alleviate the computing, storage and bandwidth pressure on the network [1,2]. For example, edge computing can significantly benefit IoT-related services by using the computing ability of local equipment to preprocess data, while edge storage can reduce network latency for video steaming services by caching popular contents in edge devices. Moreover, reducing the transfer of data traffic from edge devices to the cloud can improve privacy protection and reduce bandwidth requirements [3]. Machine Learning (ML) is a promising way to deploy more intelligence in allocating edge resources according to demands from different network services [4] [5]. The centralized control of software-defined networks (SDN) facilitates ML-based optimization for programmable management of edge computing and caching [6], and network function virtualization (NFV) [7].

The concept of intelligent edge mobile network technologies has become a focus for research in academia and industry. Many on-going research projects focus on developing scalable solutions to use the capabilities of the increasing number of smart devices and cloudlets deployed at network edges to reduce latency, increase resilience, and security, and facilitate local decision-making, e.g., the EU Horizon 2020 projects LightKone and SPOTLIGHT. The US NSF has also funded many projects in this area, including "A Knowledge-Defined Platform for Real-Time Management of Transmissions and Computations at Network Edge" and "Improving Network Security at the Network Edge. In China, the NSFC funded a significant number of machine learning and edge network related projects in 2018, for example, "Research on Green Programmable Edge Network Architecture and Intelligent Resource Optimization" [8]. This paper will investigate the theoretical models and key technologies of ML-assisted intelligent resource management for programmable mobile edge networks. To transform network management into a self-configuration, self-optimization and self-healing paradigm, which promises to guarantee various QoS requirements innovatively, we use ML approaches to support flow-based traffic engineering, then develop intelligent network resource management strategies, which dynamically allocate resources according to network environment, services' requirements and users' behaviors.

The reminder of the paper is organized as follows. Section 2 elaborates the network architecture and investigates a traffic engineer which is used to support flow-based network virtualization in edge nodes. Section 3 proposes the Natural policy gradient-based actor critic reinforcement learning algorithm to learn the energy-oriented optimal caching policy. Numerical simulations are conducted to show that our proposed cache policy can save energy consumption and improve network performance significantly in Section 4. Finally, we conclude the study in Section 5.

## 2 Mobile Edge Network Scenario Analysis and Resource Virtualization

### *2.1 Scenario Analysis*

Fig. 1 provides an overview of mobile edge networks with intelligent using of edge 3C resources. The basic idea of edge storage and computing is to move the computing and storage capacity from the cloud to the network edge devices which are closer to end users, so that users' QoE can be improved by reducing the need for backhaul bandwidth and decreasing the delivery time, moreover, reducing the transfer of data traffic from edge devices to the cloud can save energy consumption significant. For example, edge computing can significantly benefit IoT-related services by using the computing ability of local equipment to preprocess data, while edge storage can reduce network latency for video steaming services by caching popular contents in edge devices. However, many network performance-sensitive but promising applications, like smart farming, driverless cars and smart industry, are constrained by the limited edge network resources.

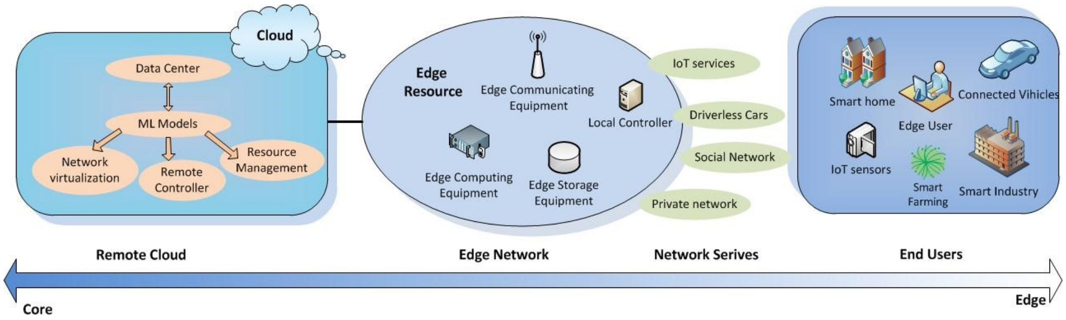

**Figure 1:** Scenario of a collaboration cloud and edge cellular through machine learning algorithms

Thus, the optimization of edge nodes deployment and their cooperation with the cloud for efficient storage and computing should be investigated. In our paradigm, the computationally intensive machine learning process of model training is accomplished in the cloud. The trained model is deployed in mobile edge nodes to decide whether and where to cache data locally or to forward it to the cloud. In this paper, we focus on how to cache popular contents properly with limited edge storage, so as to reduce energy consumption in mobile edge network. Besides, we virtualize edge network infrastructure to facilitate more flexible edge resource allocation for different network service providers.

### *2.2 Intelligent Mobile Edge Network Virtualization with 3C resources*

Making the best use of the computing, caching and communication (3C) resources of mobile edge networks can achieve better network performance and better use QoE with lower energy cost, thus end users, service providers (SPs) and infrastructure providers (InPs) will all benefit. However, traditional strategies for provisioning mobile networks require forecasts of user demand, and pre-allocation of 3C resources to anticipate that demand. Service provider resources will often be wasted, or user experience will be poor, if the live demand varies from forecasts. Service providers will incorporate headroom into their resource requests to InPs. These mechanisms all waste resources. Fortunately, the centralized control of software-defined networks (SDN), which splits the function of network device into control plane and data plane, provides a flexible network management that benefits edge computing and caching [6]. Fueled by network function virtualization (NFV), which virtualizes the function of network device on general computer hardware, the caching or storage, computing and communication functionalities are beginning to converge [7].

This paper proposes a paradigm which allows InPs to offer virtual networks (VMs) to SPs. The aggregate demand from SPs will be relatively static, because of the large user population being supported. Thus the aggregate demand on InP resources changes slowly and it can focus on satisfying changing SP demands through balanced 3C resource allocation across VMs to meet performance objectives at the lowest cost. We place a virtualization layer, mobile virtual network operators (MVNO), siting between the underlying physical edge access network and virtual wireless networks. The physical network consists of various infrastructures such as base stations (BSs) and relay nodes (RNs), while virtual wireless networks consist of logical controllers and programmable switches. The virtualization layer acts as a specialized controller, physical resources are abstracted as virtual resources and reallocated to each VM according to isolation mechanisms which is controlled by virtualization layer. For example, virtualization of the wireless medium can be achieved in multiple ways such as space division multiplexing, frequency division multiplexing, coding division multiplexing, time division multiplexing, hybrid approaches, etc. At the same time, mobile edge networks separate control from the data path through an open API like OpenFlow, and protocols like FlowVisor are used to create virtual network slices and provide network resource isolation among them. Correspondingly, with this networking architecture, SPs just need to declare its requirement to virtualization layer and concentrate on providing services to its subscribers based on the VM, without considering the diversity of infrastructure, meanwhile, InP who owns the underlying physical resource just need to allocate resource to meet demand of virtualization layer most efficiently.

### *2.3 Traffic Flow Processing*

In order to create virtual network slices, we suggest a reliable and real-time traffic flow monitoring mechanism in the physical layer. Since the rate of the data packets pass through a device can be very fast, it is hard to detect and process all the data packets. How to make efficient use of the limited edge storage and computing resources should be investigated. As shown in Fig. 2, to reduce the caching and computing pressure of edge devices, only a small percentage of data plane packets will be sampled. The sampled packets are delivered to control plane, then divided into training, validation and test datasets to generate the trained model. The trained model can be deployed in the edge nodes for prompt and accurate processing locally at the edge for new incoming packets [9]. Edge nodes have the ability to do minor adjustment of the trained model parameters locally adapting to the constant changing environment, to reduce the frequency of requesting full retraining of the model in the cloud. Extended OpenFlow protocol is used to enable a logical controller to maintain global traffic monitoring for better support of virtual resource management. Complex caching and computing tasks are handled by a remote logical controller, while simple processing tasks that can be accomplished with limited caching and computing resources can be locally dealt with by edges nodes in the data plane.

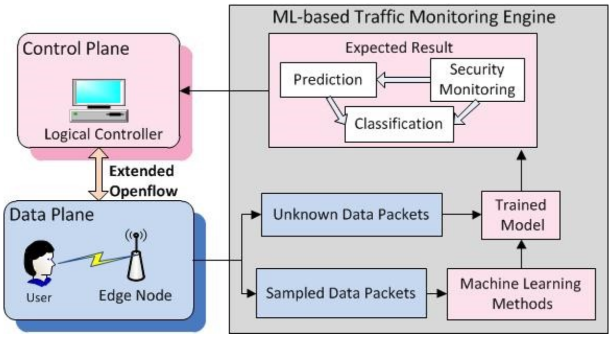

**Figure 2:** Concept of the flow-based traffic processing engine

Most edge devices do not have enough resource to perform computation intensive Deep Packet Inspection (DPI) for every data packet [10]. So semi-supervised ML-based approaches can be used to realize a fine-grained and real-time classification engine with low computation cost at network edges. In addition, DPI cannot inspect encrypted packets [11]. In order to make the best use of unlabeled data packets and improve the performance of trained model, semi-supervised learning methods, which use both labeled and unlabeled data, are applied to obtain an efficient way of using edge computing resources. The outcome from this processing engine can be used to support a real-time programmable network architecture. Traffic features can be extracted from the classified packets for flow-level traffic prediction.

On the other hand, adjusting resource allocation with traffic prediction is an effective way to save network resource cost. We implement an accurate traffic prediction mechanism for software defined access networks. Traffic flow prediction heavily depends on historical data traffic, thus traffic features are required in not only time dimension but also space dimension [12]. In this case, supervised learning systems which have clear corresponding output variables based on labeled training datasets are appropriate. We adapt appropriate ML algorithms according to statistic characteristics of network traffic. The main purpose of the predictions is to pre-adjust energy state, allocate resources and cache popular contents in edge devices. Prediction can also be used for network abnormality detection, for example, alert can be generated when network traffic is significantly different from prediction.

A flow-granularity anomaly-based intrusion detection system is also required. Unsupervised machine learning algorithms can be used to avoid resource waste on anomaly traffic, which is apt at operating over streaming data and able to detect new types of attacks including zero-day attack. Intrusion detection can be considered as a special traffic classification, in which the sources of attacks and features of anomaly traffic are unknown. Unsupervised learning methods assume unknown class labels and cluster cases together based on some measures of similarity [13]. Thus, a concise and accurate representation for statistics of network flow metrics can be created to address the high-dimensional problem of data streams. An SDN architecture is able to extend clustering algorithms for classifying normal and abnormal network data streams. The proposed anomaly detection engine can be applied to real application traffic to evaluate its performance such as detection accuracy, delay, false alarm rate, etc.

## 3 Intelligent Resource Allocation

### *3.1 Problem Analysis*

With the global network view, it is possible to implement real-time resource allocation policies according to network conditions and user requirements, however, because of users' mobility, extracting characteristics from mobile users' behavior is much more difficult than from network service traffic flow. In the proposed paradigm, InPs can offer virtual networks to SPs through MVNO. InPs can thus focus on satisfying fixed SP demands with the lowest 3C resources cost, so InPs, SPs and edge users can all benefit. To achieve this, Reinforcement Learning (RL) algorithm is applied.

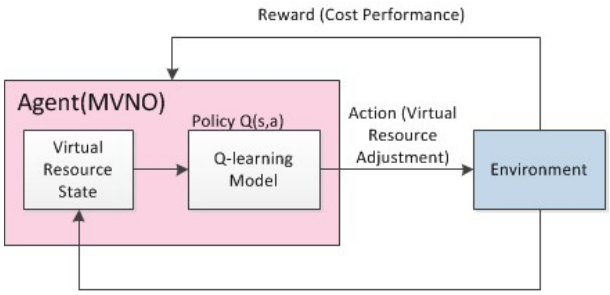

**Figure 3:** Concept of the self-adaptive resource allocation through reinforcement learning

As shown in Fig. 4, we formulate network virtualization and virtual resources allocation problems as a model-free RL framework under unknown mobile network environment. The framework aims at energy efficient resource allocation. We use MVNO as an agent to take an action according to the current virtual resource state, and then use the cost performance as a reward to assess the actions. The objective of the agent is to learn the optimal resource allocation policy to maximize a long-term reward in dynamic networks. The logically centralized control of SDN and resource virtualization of NFV facilitates energy efficient resource allocation. In order to choose the highest cost-performance resource allocation strategy in real scenario, we investigate the statistical property of resource state in wireless access network devices, then research the dynamics of network environment and service requests. We will use Markov process to model virtual state transitions, and design a RL-based control system, to intelligently adjust the 3C resource allocation policy by leaning the network environment.

### *3.2 System Model*

In order to intelligently embed virtual networks with the synergy of caching, computing and communications, network slicing protocols will be extended to support computing and storage resource isolation, so as to offer virtual resources to SPs [14]. We consider a cellular wireless network consisting of a BS B, which can connect any equipment in the cellular, and a set $R=\{R_1,R_2,\dots,R_m\}$ of RNs, each RN is directly connected to the BS. dmax denotes the maximum service radius of an RN, an RN Rm has a set $SB_m=\{SB_{m,1},\ SB_{m,2}\dots,SB_{m,L}\}$ of storage blocks, each of which is able to cache a content. There are a set $U=\{U_1,U_2\dots,U_N\}$ of users, each of which access the Internet through RN or BS directly. There are a set $VM=\{VM_1,VM_2\dots,VM_K\}$ of virtual networks, each of which belongs to a service provider. In this paper, we assume that each user require a content from each SP during t *th* time slot , and the content is different from which is delivered to the user in (t − 1) th time slot, so there are $N\times K$ contents delivered to users in total through mobile edge network. A storagelimited edge network is considered, this can be presented as $M\times L\geq N\times K$.

We use Cn,k(t) to denote the content of user Un requires from service provider V Mk at time t. According to the proposed paradigm, the popularity of a content is related to not only when it is required but also who requires it and who offers it. We use λ1, λ2, λ3 to denote the probability of content Cn,k(t), which is from V Mk and required by user Un, is not cached in edge nodes, cached in an edge node as Cx,k(t − 1) (where x 6 = n), and cached in an edge node as Cn,y(t − 1) (where y 6 = k).

Thus, the power consumption Ptotal for delivering all the contents in a edge network is modeled as

$$P_{total}=NK\left[\lambda_1\left(P_{cloud}+P_{edge}\right)+(\lambda_2+\lambda_3)P_{edge}\right] \tag{1}$$

where $P_{edge}$ and $P_{cloud}$ separately denote the power consumption of delivering a content from cloud to edge network and from edge network to user equipment. In this $P_{cloud}$ can be considered as a constant in this situation, thus it is obvious that the higher cache ratio in a mobile edge network, the lower transmitting energy will be consumed.

### *3.3 Problem Formulation*

Many cache policies can be used for enhancing cache ratio, such as caching content according to its publication time or popularity. In this paper, we assume that all the required contents has been cached in storage blocks in edge nodes, since the cache necessity of a content is related to its user and network service, we focus on where to cache a content in a mobile edge network. We use Cn,k (t) = SBm,l to describe content Cn,k, which is from virtual network V Mk to user Un, is cached in storage block SBm,l on RN Rm at t th time slot.

The energy consumption of a cached Cn,k content from edge nodes Rn to user equipment Um can be written as

$$P_{egde}^{C_n,k} = \begin{cases} P_{R_{m,U\_n}}, & d_{R_{m,U_n}} \leq d_{max} \\ P_{R_{m,B}} + P_{B,U_m}, & d_{R_{m,U_n}} > d_{max} \end{cases} \tag{2}$$

where dRm,Un denotes the distance between RN and user equipment. When the content required by a user is cached in an RN which can connect to the user, the content will be delivered to user directly, otherwise, the content will be submitted to BS, then delivered to user.

The energy efficiency optimization problem can be formulated as

$$\min \sum_{n=1}^{N} \sum_{k=1}^{K} P_{edge}^{C_n,k} \tag{3}$$

Under an edge storage space constraint, we formulate this problem with a reinforcement learning model, the state space is defined as S, system state st is therefore a vector defined as:

$$s_t = \{C_{n,k}(t)\} \qquad \text{where } \forall U_n \in U; \forall VM_k \in VM \tag{4}$$

In our problem, the agent will make decision for the requested contents at (t − 1) time slot, we define actions A = {at (Cn,k)} for the content Cn,k as whether the content will be exchanged with another. The immediate system reward will be defined as r(st,at), where r is the reward function for energy consumption saving. In this paper, we use Q-learning to adjust network statement according to network environment, the reward can be written as

$$r(s_t, a_t) = \begin{cases} R, & if\ s_{t+1} = s_{term}\ , \\ 0, & otherwise. \end{cases} \tag{5}$$

where R is a positive constant, sterm denotes that each content is cached into the best storage block.

This algorithm learns experience with the *Q* value iteration update which can be formed as

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})] \tag{6}$$

where α is the learning rate (0 < α ≤ 1) and γ is the discount factor (0 < γ ≤ 1).

By learning the system state for each action, each content will be cached into the best storage block, so as to minimize the energy consumption in a mobile edge network.

## 4 Simulation

In this work we simulate a resource allocation policy based on the network system which has been formulated in Section 3. We use a heterogenous edge network with 20 users, M (M = 2,3,4,5,6) RNs and a BS. We set the location of BS in the center of a polar coordinate, the m *th* RN is placed at $[500, \frac{2\pi m}{M}]$. The network system simulation parameters are obtained from [15]. The energy consumption of RN is assumed to be 10% of a BS. We consider an OMDMA system where the same time and frequency resources are allocated. we assume that each RN has one cache block, and serve one user in a time slot. Each user randomly requests a content from a virtual network. The number of virtual network equals to user number. We set the content size as 500KB, which should be concerned is, the content size does not affect energy efficiency but affect transfer time. The pass loss between edge nodes X and Y is denoted as PLX,Y , according to [16], the calculated path losses are summarized as PLB, R = 11.7 + 37.6lg(d), PLR, B = 42.1 + 27lg(d), PLR, R = 38.5 + 27lg(d) and PLR, U = 30.6 + 36.7lg(d). The algorithm parameters are set as α = 0.2, γ = 0.8, R = 100.

Most of the researches about edge cache [4,17,18] are limited in prediction- based cache policies without use of edge communication ability. For comparison, we simulate an edge network without cache ability, with prediction-based cache policy and with our proposed Q-learning cache policy under same benchmark setting. In a non-cache edge network, RNs send contents which are collected from local sensors to cloud through BS, a content will be sent back to edge if an edge user makes a request. In an edge network with static cache policy, the contents are cached in edge storage according to request prediction which is obtained by analyzing content popularity and user behavior, if a requested content is not stored in proper edge nodes, the content still need to be acquired from far cloud.

Fig. 4 shows the network energy efficiency as a function of relay numbers, which determine the edge communication and cache abilities. By comparing the different edge networks, the edge nodes with storage can significantly improve network energy efficiency. The Q-learning cache policy is more energy efficiency than prediction-based cache policy as the number of relays growth when M > 2. What calls for special attention is that the Q-learning cache policy has no advantage when the edge nodes are not enough to composing an edge network

(M = 2 and M = 3).

In consideration of the other network performance metrics, the time of delivering all the contents according to users' requests and the throughput are analyzed. To immune the result from core network factors, we just consider the users who are served by edge nodes. Fig. 5 and Fig. 6 show the delivering time and throughput as functions of cache hit ratio, which denotes the probability that a user's requested content has been cached in a proper node. We can see that with the cache hit ratio growth, the delivering time linearly decreases and the throughput increases. What calls for special attention is that the more relay nodes in an edge network, the more delivering time is required, this may result in lower QoS. Thus a trade-off between energy consumption and delivering speed should be considered. However, there is no obvious difference on throughput.

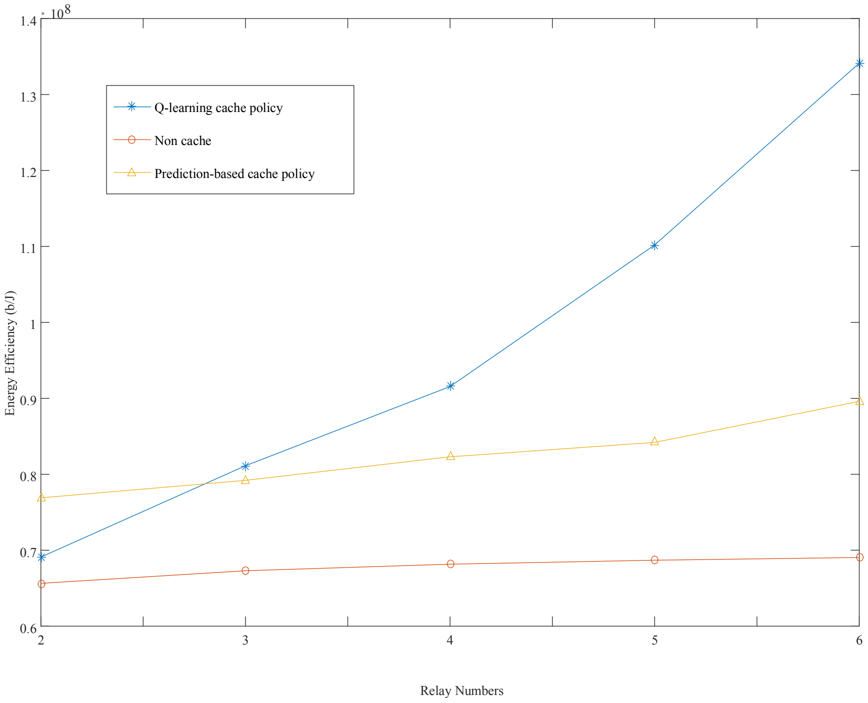

**Figure 4:** Network energy efficiency as a function of relay numbers

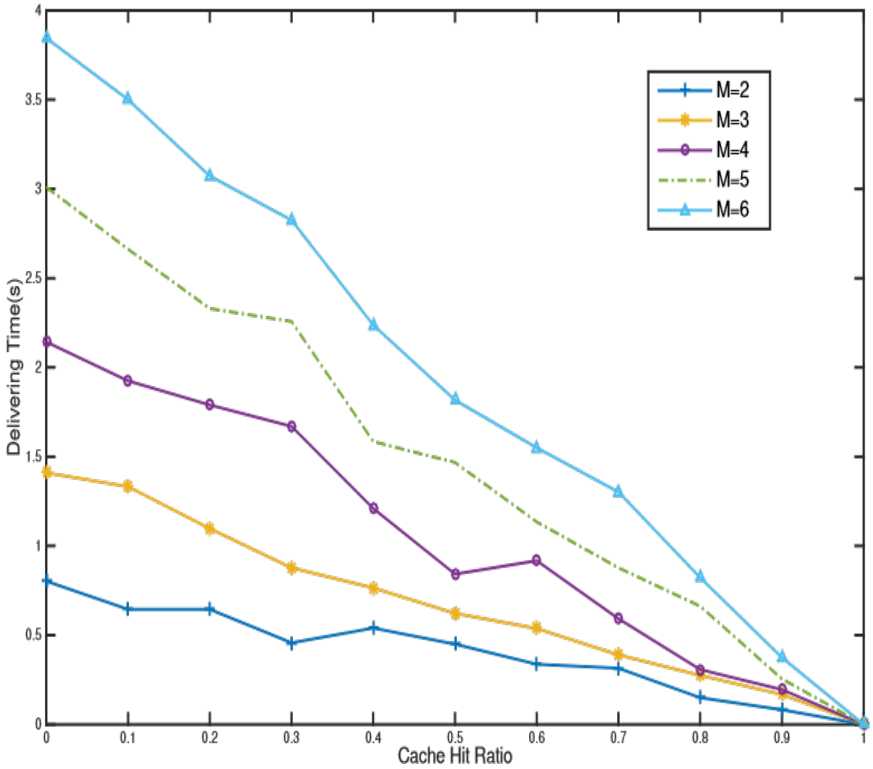

**Figure 5:** Delivering time as function of cache hit ratio for different RN numbers

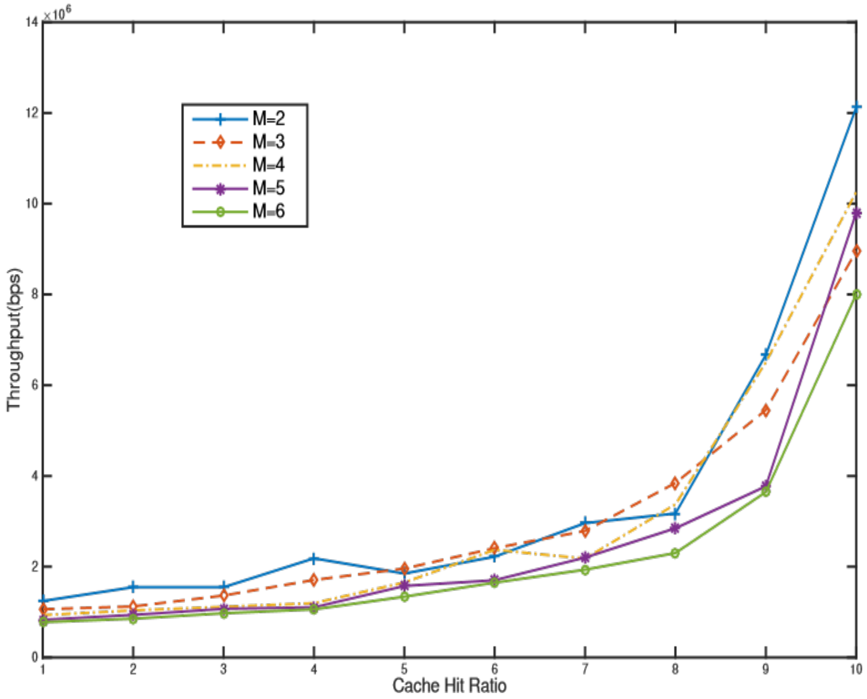

**Figure 6:** Throughput as function of cache hit ratio for different RN numbers

## 5 Conclusion

This paper aims at investigating the intelligent use of edge devices' caching and computing capabilities in network resource management by using machine learning-based policies. We design a paradigm of programmable mobile edge network with communicating, caching and computing abilities. Then suggest a flow-based traffic engine to realize network virtualization in an efficient way by processing data traffic through machine learning approaches. Then we apply reinforcement learning for dynamic virtual network resource allocation, so as to reduce the energy consumption and improve network performance. Although it may not be scalable enough to accommodate all the edge networks in the Internet, it can be applied in a smaller scope and still benefit everyone. We are currently prototyping this scheme and evaluating its performance in an experimental testbed. Our goal is to find the optimal parameter settings for the current Internet traffic patterns within the constraints of the current technologies and develop a systematic way to evolve the scheme as traffic patterns and technologies progress.

**Funding Statement:** This work was supported by the National Natural Science Foundation of China (61871058) and Key Special Project in Intergovernmental International Scientific and Technological Innovation Cooperation of National Key Research and Development Program (2017YFE0118600).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, no. 5, pp. 637–646, 2016.

[2] D. Liu, B. Chen, C. Yang and A. F. Molisch, "Caching at the wireless edge: design aspects, challenges, and future directions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 22–28, 2016.

[3] E. Bastug, M. Bennis and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, 2014.

[4] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393–430, 2018.

[5] L. V. Le, B. S. Lin and S. Do, "Applying big data, machine learning, and SDN/NFV for 5G early-stage traffic classification and network QoS control," *Transactions on Networks and Communications*, vol. 6, no. 2, pp. 36–36, 2018.

[6] A. C. Baktir, A. Ozgovde and C. Ersoy, "How can edge computing benefit from software defined networking: A survey, use cases, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2359–2391, 2017.

[7] N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.

[8] Natural Science Foundation of China (NSFC) Project, "Research on green programmable edge network architecture and intelligent resource optimization". https://isisn.nsfc.gov.cn/egrantindex/funcindex/prjsearch-list#.

[9] M. Uddin and T. Nadeem, "TrafficVision: A case for pushing software defined networks to wireless edges," in *IEEE 13th Int. Conf. on Mobile Ad Hoc and Sensor Systems*, pp. 37–46, IEEE, Brasilia, 2016.

[10] P. Wang, S. C. Lin and M. Luo, "A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs," in *IEEE Int. Conf. on Services Computing*, pp. 760–765. IEEE, San Francisco, CA, 2016.

[11] I. Orsolic, D. Pevec, M. Suznjevic and L. Skorin-Kapov, "A machine learning approach to classifying YouTube QoE based on encrypted network traffic," *Multimedia Tools and Applications*, vol. 76, no. 21, pp. 22267–22301, 2017.

[12] Y. Lv, Y. Duan, W. Kang, Z. Li and F. Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[13] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[14] Y. He, N. Zhao and H. Yin, “Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 44–55, 2017.

[15] L. Li, Y. Wei, M. Song and X. Wang, “Flow-table updating strategy for efficient use of renewable energy in software defined wireless relay networks,” *Journal of Communications and Networks*, vol. 19, no. 6, pp. 605–617, 2017.

[16] A. B. Saleh, S. Redana, J. Hämäläinen and B. Raaf, “On the coverage extension and capacity enhancement of in band relay deployments in LTE-Advanced networks,” *Journal of Electrical and Computer Engineering*, vol. 12, 2010.

[17] W. X. Liu, J. Zhang, Z. W. Liang, L. X. Peng and J. Cai,” Content popularity prediction and caching for ICN: A deep learning approach with SDN,” *IEEE Access*, vol. 6, pp. 5075–5089, 2017.

[18] G. Zhang, Y. Li and T. Lin, “Caching in information centric networking: A survey,” *Computer Networks*, vol. 57, no. 16, pp. 3128–3141, 2013.