Smart Resource Allocation for Mobile Edge Computing: A Deep Reinforcement Learning Approach

Pavitra Basappa Gudimani

Summer term 2020

Abstract

The amount of data generated by all the mobile devices in the future can cause congestion at the cloud for data processing and cause a delay in service time. Mobile Edge Computing (MEC) architecture extension of existing cloud computing is promising solution. MEC improves the response time by moving network and computing resources closer to the customer. MEC does not send the data to the cloud for further processing rather the edge network itself will process the data. The burst request at MEC faces an issue in allocating computing and network resources. At this stage, Deep Reinforcement Learning based Resource Allocation scheme adapts to a mutative nature and allocates computing and network resources. Thus, it reduces the service time to users and offers fairness in the use of resources at MEC. DRLRA achieved better results in varying MEC environments than the traditional OSPF algorithm.

1 Introduction

The phenomena of embedding the sensors in smart devices to sense the environment have changed the way of living life. The data sent by IoT devices from smart cities, smart healthcare, smart vehicles [1], etc. results in a high volume of data. And it is produced at a high pace (the European Commission has predicted that there will be 50 to 100 billion smart devices connected to the Internet by 2020) [2]. Also, Cisco has predicted that the devices connected to the Internet will generate 507.5 ZB/year by 2019 [2]. The generated data is further transferred to the remote cloud. The volume and speed of the data generated create congestion at the cloud. Thus, efficient data processing architecture is required.

At this stage, Mobile Edge Computing architecture aid in reducing the network congestion and response delay. It deploys the applications in a distributed form in mobile edge servers to deliver the computational and critical applications services. It appears to be an encouraging solution to the problem faced by several mobile devices. On the other hand, it has following limitations. Primarily, the initial cost of investment to deploy MEC servers(MECS) at the base stations and for their maintenance is expensive. Secondly, Only a restricted number of applications can be deployed to supply services to innumerable edge devices [2]. Lastly, the dynamic nature of human life brings new requirements to the devices, continuously changing the location of requests generation, and inconsistent quantity of data, all these factors cause serious burst over MECS, increases the computational load, and congestion in the network links of certain regions.

MEC fails to provide flexibility in granting computing and network resources. Therefore, the author has presented DRLRA algorithm to assign computing and network resources accordingly under the dynamic circumstances of MEC. This paper illustrates following

- DRL Technology [3] understands the dynamic behavior of the MEC, and generates a sequence of decisions.
- Merging Software Defined Network technol-

ogy [4] with MEC architecture centralizes the distributed network.
- The goal is to minimize the service time (Computing + Routing delay) by utilizing DRLRA.

2 SDN-enabled MEC architecture

Deploying all the applications on a single MEC Server is expensive. MECS solves the problem by virtualizing the resources. When mobile requests for an application, it will be sent to MECS in its vicinity. If there is no required application, the request will be routed to other MECSs with the needed application deployed. Hence, the total service time includes the access time to nearby MECS, routing time among MECSs, and processing time at corresponding MECS.

To have the request delivered to destined MECS in the mutative environment the routing should happen correctly. That's where the emerging SDN technology comes in handy to understand the state of the network. In this context SDN uses below planes [5].

1. Data Plane - Forwarding devices Switches, routers, and gateways reside in this plane and work according to decisions made by the controller in the control plane.
2. Control Plane - The core of SDN architecture and all forwarding decisions made upon the control logic provided here.

DRL unit with the DRLRA algorithm is installed in the SDN controller plane and it takes dynamic states of the network as input from the data plane and it trains the agent to make decisions, based on which SDN controller can perform forwarding.

3 Problem Definition

MEC without resource allocation algorithm faces below problems

Edge network routing delay: The routing delay for the aggregated requests of mobile devices at nearby MECS to the MECS that is hosting the required application. [6].

Data Processing delay: Number of CPU Cycles required for computation at the destined MECS.[6].

Variance of network resource allocation: Several requests will be forwarded to destined MECS. Allocation of network resources on the data links should be stabilized to reduce network traffic and to transfer the request.

Variance of computing resource allocation: The destined MECS is responsible for processing the received requests. Different requests follow various routing path to reach destined MECS and cause a difference in computing load at each MECS.

4 Deep Reinforcement Learning Based Resource Allocation Algorithm

A smart resource algorithm understands the varying environment of the MECS and makes a feasible sequence of decision on resource allocation using DQN framework.

4.1 Reinforcement Learning

It is all about training an agent without labeled data that communicates with the environment. An agent performs an action in a state and grabs the reward. The sole purpose of an agent is to maximize the total rewards over the episode. By giving positive rewards, an agent can strengthen itself to develop a strategy or policy [7]. In the given situation, the agent interacts constantly with the MEC environment, makes a sequence of decisions to get corresponding rewards [6].

- **State (s):** MECS environment with gathered requests.
- **Action (a):** Action performed by MECS to fulfill the user request.
- **Episode (k):** Sequence of states, actions, and rewards that occurs between initial state and termination state. In this context, it is a resource allocation process from MECS where requests from mobile devices gathered to get a service of application to destined MECS where the required application is installed.
- **Policy pi:** is mapping from states of the environment to actions to be taken in those states [8].

- **Reward (r):** is positive feedback given, when an agent performs good action.
- **Cumulative Reward:** Reward obtained by an agent at the end of each episode.
- **Action value function (Q function):** The value function is to quantify the quality of the action made at the state.

4.2 Deep Q Network

DQN is a powerful framework of DRL. It takes the current state as an input and gives action value for every action that can be taken in that state i.e Q(s,a;ω), where ω is a parameter of network. Action with the highest Q value $a = argmaxa_tQ(s_t, a_t; \omega)$ is selected using ϵ-greedy strategy. To make the generated Q value closer to the target Q value, Mean Square Error, a loss optimizing function is used [6].

4.3 Training

DRL agent in a MEC environment is trained using DQN framework. Figure 1 explains the training process. Experience Replay memory is installed to store environment response for taken action and parameters ω and ω^- are set for target and evaluation network respectively. Episode k is initialized with state s_t at time t. s_t is taken as input to evaluation network and using the ϵ-greedy method action a_t with maximum Q-value is chosen. The agent gets a reward r_t and next state s_{t+1}. $Q(s_t, a_t, r_t, s_{t+1})$ is stored in experience replay. To train the network, sample state s_t and action a_t applied to evaluation network from memory. $Q(s_j, a_j; \omega)$ and $\gamma maxa_{j+1}Q'(s_{j+1}, a_{j+1}; \omega)$ calculated from evaluation and target network respectively. The loss function is used to bring the Q value of evaluation network closer to the target network through gradient descent and backpropagation. Finally, the trained evaluation network is obtained after performing resource allocation of t steps in episode k [6].

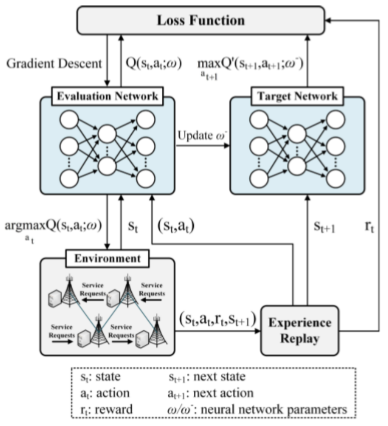

Figure 1: Detailed DQN Framework[6]

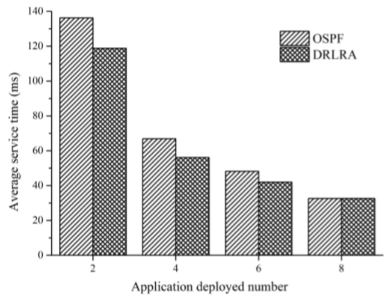

Figure 2: OSPF and DRLRA comparison under different application deployed number [6]

5 Performance Evaluation

Average service time minimization: Fig.2 shows that the average service time decrease with an increase in the number of deployed applications. As the number of deployed application increase, each MECS will almost have the requested application. Therefore, the distance between the location of requests generated for a specific application and MECS hosting the required application will reduce and thus result in a reduction in average service time. At a certain point, when the number of deployed applications reaches a maximum, the service time is considerably decreased. There are also other parameters considered for the comparison between DRLRA and OSPF [6].

Resource Allocation Balancing Analysis: Fig. 3 depicts that OSPF performance degrades over the increase in the complexity of the MECS environment by rising number of deployed applications. On the other hand, DRLRA, while making decisions on path selection for routing it learns the situation and adapts to it. Therefore, DRLRA allocates resources

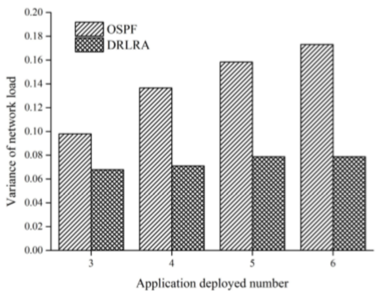

Figure 3: OSPF and DRLRA comparison under variance of network load[6]

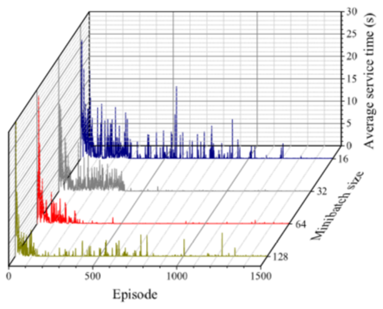

Figure 4: Convergence performance of DRLRA[6]

by using it evenly and prevents network congestion.

Convergence Performance Analysis: Fig.4 shows as the batch of samples provided increases, the speed required to stable the average service time is reduced. The agent will not get a chance to learn enough with the small number of samples. With the huge batch size agent might converge to the poor average service time. Therefore, the size of samples should be chosen carefully.

Conclusion

This paper included the serious problem of assigning computing and network resources when the request burst is in high volume. The proposed solution is, MEC architecture with DRL to reduce the average service time. The DQN based DRLRA algorithm's capacity to understand the mutative environment makes itself fit for the MECS environment. DRLRA is compared with OSPF algorithm on different experiments, the output has proven that DRLRA is more effective in the varying environment. Further, it can be improved with Double DQN [7] and by providing more environmental details.

References

[1] Wen Sun, JiaJia Liu, and Haibin ZHang. *When Smart WearableS meet IntellIgent VehIcleS: challengeS and Future dIrectIonS*. IEEE Wireless Communications, 2017.

[2] Xiang Sun and Nirwan Ansari. *EdgeIoT: Mobile Edge Computing for the Internet of Things*. IEEE Communications Magazine, 2016.

[3] *Reinforcement Learning*. URL: `https://en.wikipedia.org/wiki/Reinforcement_learning`.

[4] *Software Defined Network*. URL: `https://en.wikipedia.org/wiki/Software-defined_networking`.

[5] Kuljeet Kaur et al. *Edge Computing in the Industrial Internet of Things Environment: Software-DefinedNetworks-Based Edge-Cloud Interplay*. IEEE Communications Magazine, 2018.

[6] Jiadai Wang et al. *Smart Resource Allocation for Mobile Edge Computing: A Deep Reinforcement Learning Approach*. IEEE Transactions on Emerging Topics in Computing, 2019.

[7] *Deep Q Networks*. URL: `https://towardsdatascience.com/qrash-course-deep-q-networks-from-the-ground-up-1bbda41d3677`.

[8] *Policy*. URL: `https://stackoverflow.com/questions/46260775/what-is-a-policy-in-reinforcement-learning`.