Optimized Load Balancing Mobile Network using a Generative Adversarial Network Based Network Simulator

Load Balancing Mobile Network by GAN

Tin-Yu Wu
Department of Computer Science and Information Engineering
National Ilan University
Yilan County, Taiwan
tyw@niu.edu.tw

Yueh Wu
Department of Computer Science and Information Engineering
National Ilan University
Yilan County, Taiwan
willy821230ptr@hotmail.com.tw

Fu Jie Tey
Department of Computer Science and Information Engineering
National Ilan University
Yilan County, Taiwan
alexsword88@gmail.com
line 4: e-mail: name@xyz.com

Bo-Hong Huang
Department of Computer Science and Information Engineering
National Ilan University
Yilan County, Taiwan
b0643027@ems.niu.edu.tw

***Abstract*—With the advances of neural networks and the 5th generation mobile networks (5G), how to use Artificial Intelligence (AI) in 5G wireless networks has become a widely discussed topic while neural network is one form of artificial intelligence. Due to the complexity of 5G networks, it would be difficult to achieve load balancing. For this reason, before the 5G networks are officially launched, this study would like to investigate the processing capacity and learning capacity of neural networks over complicated problems. We combine Generative Adversarial Network (GAN) with the network simulator ns-3 and use neural networks for load-balancing simulation parameter adjustment and load balancing optimization.**

Keywords- 5G; GAN (Generative Adversarial Network); load balance; neural network

I. INTRODUCTION

When a world of 5th Generation Mobile Networks (5G) is approaching, we are expecting a complex environment interwoven with mobile network, Wi-Fi, millimeter wave and so on. Many methods have been proposed to solve the load-balancing problems for wireless networks, but most of them focus on the switch between Wi-Fi, D2D (Device to Device) and large/small base stations (BS).

On the edge of large-scale BS coverage, the signal strength is poor and the interference is significant. Additional small BSs can be deployed to fix the problem but it requires deployment cost. Therefore, considerations must be made to ensure if the existing BSs can possibly achieve load balancing. With the rise of Software-Defined Networking (SDN), the load balancing methods for wire-line networks have been extensively studied and become quite mature. However, it is inappropriate to use the methods for wire-line networks in wireless networks. In wire-line networks, nodes are connected by physical wires. According to SDN, a centralized controller gathers load balancing data and directs the data flow to the chosen destination to achieve load balancing. However, compared with wire-line networks, the transmission efficiency of wireless networks is far lower because of the path loss and interference, making the load balancing methods for wire-line networks not suitable for wireless networks. Therefore, this study attempts to make use of the neural networks to enable load balancing in wireless networks.

A network simulator is required for simulation of load balancing and it could be somebody else's design, like ns-3, or a simulator built according to one’s needs. In this study, ns-3 is adopted. During the simulation, we will setup parameters, conduct the simulation, investigate the results, analyze the performance and adjust the parameters for follow-up simulations. Because such a series of steps must be repeated and requires a lot of time, we hope that by integrating neural networks with simulation systems, the parameters can be automatically adjusted and the target parameters can be figured out without substantial modifications of the overall simulation environment. We will use Generative Adversarial Network (GAN): utilizing its parameters and combining GAN with the simulation system to achieve automatic generation and parameter adjustment [11][12].

The rest of this paper is organized as follows: Section II reviews researches on load-balancing methods and background. Section III details how about system architecture and proposed method. In Section IV, we perform the experiment and simulate the parameters with our proposed models. Finally, we summarize our findings and results.

II. BACKGROUND AND RELATED WORKS

This section will introduce load-balancing methods, Generative Adversarial Network (GAN) and SeqGAN.

ISBN: 978-1-61208-758-0

A. Load Balancing Methods

In wire-line networks, a centralized controller is responsible for monitoring computers and routers in the network, and managing the data flow to achieve load balancing.

- Standard Deviation Based

The standard deviation based load balancing algorithm depends on the centralized controller to collect load-balancing data of all routers and compute the standard deviation of each router. The standard deviation is usually used to measure the spread of the data about the mean value. For load balancing, the standard deviation is used to find out a light-loaded router so that flows of a congested path can be redirected to a light-loaded path [1].

- Weight Based

In the weight based load balancing algorithm, all routers are given assigned weights initially. According to the load-balancing status of the current BS, weights can be adjusted accordingly [2].

- Mobile Network and Wireless Network

Due to the reasons mentioned above, the load-balancing algorithms designed for wire-line networks are not suitable for wireless networks. But, many methods have been proposed to implement load balancing. Most of them propose to transfer mobile devices to the neighboring BSs to alleviate congestion [3], to use Device-to-Device (D2D) communications to offload the data traffic in the congested small cell to the neighboring small cell BSs [4], or to use an architecture based on fog computing [5]. Note that D2D communications are not allowed under all circumstances. Still some are based on Wi-Fi to determine how and when to connect with the right AP [2] [6]. Nevertheless, Wi-Fi and cellular networks are different in signal strength, interference and so on. Even earlier, some algorithms have been proposed for mobile load balancing in LTE system [7]. However, the existing simulation environment architectures are not suitable for the new 5G networks.

B. Generative Adversarial Network (GAN)

Generative Adversarial Network (GAN), first presented in 2014, combines a generative network with a discriminative network [8]. The generative network generates new data instances and the discriminative network identifies if the data is real or fake. The two fight against each other to improve throughout the training process. Take the imaging data generated by the generative network for example. When the samples get closer to real images, the discriminative network can discriminate more and more accurately. However, if the generative network's training is too fast, the discriminative network might not be able to converge. On the contrary, if the discriminative network converges too fast, the generative network will be affected also. GAN has often proved difficult to train because it is hard to converge, or the result is not as expected. To cope with this problem, parameter adjustments together with some real data are required. The development of GAN has grown rapidly and most of them focus on discriminating images, such as DCGAN [9] and WGAN [10]. In this study, we choose SeqGAN.

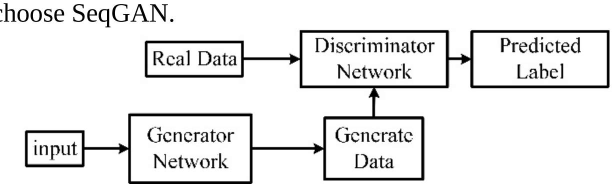

Figure 1. Generative Adversarial Network

C. SeqGAN

Proposed in 2016, Sequence GAN (SeqGAN) [11] enables GAN to process discrete outputs. GAN can differentiate sequential image data but not languages or words. To solve this differentiation problem, SeqGAN directly performs gradient policy update. The RL reward signal comes from the GAN discriminator judged on a complete sequence, and is passed back to the intermediate state-action steps using Monte Carlo search.

III. SYSTEM ARCHITECTURE AND PROPOSED METHOD

This section introduces the design of both the proposed load-balancing environment and Generative Adversarial Network (GAN), as displayed in Figure 2. A GAN consists of two neural networks: generator and discriminator. Finally, we will describe the full process.

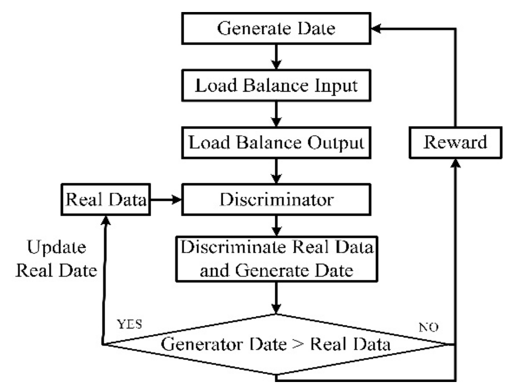

Figure 2. System Architecture

A. Load Balancing Simulation

For load balancing simulation, we first set and adjust parameters, including number of BSs, number of users, simulation time and weighting parameters defined by GAN. Simulations can be conducted using self-motivated software or specific simulators. After the simulation, we output the results for performance analysis. In this study, the result output is the load balancing test throughput.

B. Combing GAN with Load Balancing Simulation Test

To combine GAN with the load balancing simulation test, the data generated by the generator must correspond to the parameters of the network simulator. The result of the load balancing simulation will be returned to the neural network as feedback for further analysis.

- Corresponding Parameter Design

Based on SeqGAN, our proposed method generates a one-dimension array as weighted parameters for BSs in the load-balancing simulation. This study evaluates "throughput," instead of "test loss" because the test loss is defined as a percentage. It is possible that the throughput greatly varies but the test loss is similar. Therefore, the higher throughput means the better performance of the load-balancing simulation test. The result is then sent back to the neural network.

- Adjustment of SeqGAN

The generator based on SeqGAN and our proposed method is basically the same: a one-dimensional array composed by normally distributed random numbers. The discriminator then identifies if the generated data is real or fake. If the data is recognized as authentic by the discriminator, the data generated by the generator will be very close to real data. The real data of the original SeqGAN are poetry data generated by a trained RNN but its goal is different from ours. Therefore, we take the generated data of a better load balancing simulation result as the basis of real data.

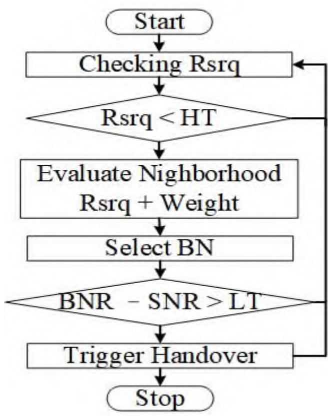

Figure 3. BN selection based on weights

- Impact of Weight Parameters on BS Selection

To investigate the impact of weight parameters on BS selection, we initially selected the BS with high weight value. However, the result could be bad if the user connects to a BS providing low signal strength. For this reason, signal strength and weight values generated by the generator are both taken into consideration while choosing the best BS. According to the flowchart shown in Figure 3, the user first checks the current Reference Signal Receiving Quality (RSRQ). If the RSRQ is smaller than High Threshold (HT), proceed to the handover evaluation, in which the Best Neighborhood (BN) is chosen according to RSRQ and weight value. To avoid edge shake, when the current Serving Neighborhood RSRQ (SNR) minus the Best Nighborhood RSRQ (BNR) is larger than the Low Threshold (LT), the handover is triggered.

IV. RESULTS AND ANALYSIS

This section will introduce the integration of Tensorflow and ns-3, analyze the Generative Adversarial Network (GAN) training process and conduct the load-balancing test to evaluate the performance of the method.

A. Integration of Tensorflow and ns-3

As for the neural network, this study uses TensorFlow. TensorFlow is a Python learning library while ns-3 is a simulator written in C++ that uses the waf commands to enable examples and tests. To integrate TensorFlow with ns-3, we modify the file in ns-3 using Python programming language, and ask the system to run the waf commands of ns-3. In this way, the two systems can be integrated.

B. GAN Training Result

This study is based on the SeqGAN of the open source machine learning TensorFlow library. Normally, we examine the test loss to check if a neural network converges. A convergent curve decreases and becomes flat. Since we modify the input, output and data type, it is necessary to evaluate the convergence. Figure 4 shows that the test loss decreases and becomes stable. It means the convergence in the training of neural networks, confirming the importance of deep learning.

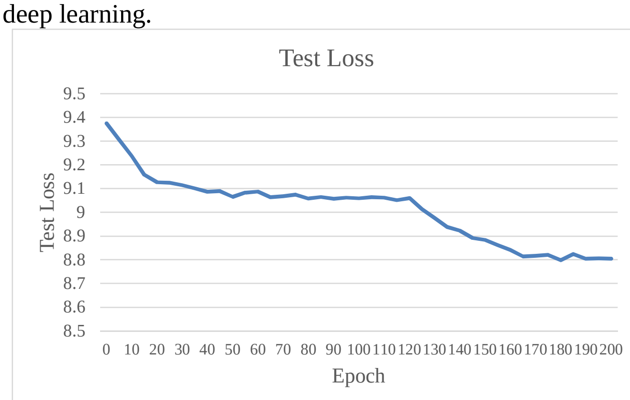

Figure 4. Test Loss

C. Load-balancing System

Instead of using self-motivated simulation software, we use the ns-3 simulator in this study. Although self-motivated software and GAN are easier, the design might not be perfect. Therefore, we use the widely recognized simulator to connect with GAN. Table 1 lists the basic load-balancing configuration parameters. Number of User Equipment (UE) is adjusted according to different modes. There are 7 BSs in the simulation and the simulation time is set to 10 seconds. Packet size and individual user throughput are determined according to the setting of the ns-3 simulator.

TABLE I. BASIC LOAD-BALANCING PARAMETERS

UE Amount	40
BS Amount	7
BS dBm	46
Simulation Time(second)	10

In the simulation scenario shown in Figure 5, BSs are deployed in an area of 1500m x 1500m. In the simulation, users are randomly distributed around BSs and weights are adjusted accordingly to achieve load balancing.

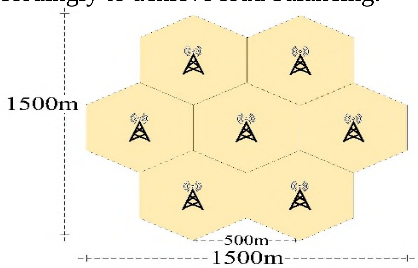

Figure 5. BS deployment in the simulation

D. *Load-Balancing Test*

In the simulation, we compare the impact of RSRQ and RSRQ Weight in which weights are generated by GAN. Figure 6 shows that RSRQ Weight obviously outperforms.

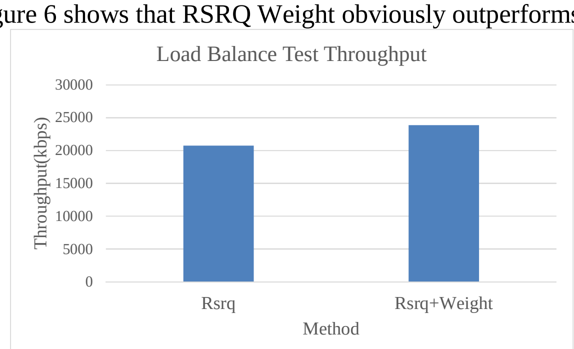

Figure 6. Throughput

V. CONCLUSION

By using a GAN based network simulator without substantial modifications, this study proposed an optimized load balancing mobile network, in which the TensorFlow learning library for Python was combined with the NS-3 simulator. The results revealed that our proposed method could reduce the packet loss rate by approximately 6% packet drop rate. Our future target is to further improve the method to build a truly competent neural network.

ACKNOWLEDGMENT

This study was supported by the National Science Council, Taiwan, under grant No. MOST 107-2221-E-197 -007 -MY3

REFERENCES

[1] Y. L. Lan, K. Wang and Y. H. Hsu, "Dynamic Load-balanced Path Optimization in SDN-based Data Center Networks," 2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), July 2016.

[2] C. Y. Lin, W. P. Tsai, M. H. Tsai and Y. Z. Cai, "Adaptive Load-balancing Scheme Through Wireless SDN-based Association Control," 2017 IEEE 31st International Conference on Advanced Information Networking and Applications(AINA), March 2017.

[3] Y. C. Wang and K. C. Chien, "A Load-aware Small-cell Management Mechanism to Support Green Communications in 5G Networks," The 27th Wireless and Optical Communications Conference (WOCC), May 2018.

[4] H. Zhang, L. Song and Y. J. Zhang, "Load Balancing for 5G Ultra-Dense Networks," IEEE Transactions on Wireless Communications, Vol.17, No. 6, pp. 4039-4050, June 2018.

[5] J. Jijin, B. C. Seet, P. H. Joo Chong and H. Jarrah, "Service Load Balancing in Fog-based 5G Radio Access Networks," IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), October 2017.

[6] K. Nahida, C. Yin and A. Zaid, "AP Load Balance Based Handover in Software Defined WiFi Systems," 2016 International Conference on Network Infrastructure and Digital Content (ICNIDC), September 2016.

[7] Y. Yang, P. Li, X. Chen and W. Wang, "A High-efficient Algorithm of Mobile Load Balancing in LTE System," 2012 IEEE Vehicular Technology Conference (VTC Fall), September 2012.

[8] R. Alec, M. Luke and C. Soumith, "Unsupervised Representations Learning With Deep Convolutional Generative Adversarial Networks," 4th International Conference on Learning Representations (ICLR 2016), May 2016.

[9] M. Arjovsky, S. Chintala and L. Bottou, "Wasserstein GAN," arXiv:1701.07875v3, December 2017.

[10] L. Yu, W. Zhang, J. Wang and Y. Yu, "SeqGAN : Sequence Generative Adversarial Nets with Policy Gradient," The Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017), February 2017.

[11] K. M. Addali, S. Younis B. Melhem, Y. Khamayseh, Z. Zhang and M. Kadoch,"Dynamic Mobility Load Balancing for 5G Small-Cell Networks Based on Utility Functions," IEEE Access, Vol. 7, pp. 126998-127011, September 2019.

[12] K. Addali and M. Kadoch,"Enhanced Mobility Load Balancing Algorithm for 5G Small Cell Networks," IEEE Canadian Conference of Electrical and Computer Engineering (CCECE2019), May 2019.