Non-uniform Time Slice Parallel Simulation Method Based on Offline Learning for IEEE 802.11ax

Yangyang Xin, Mao Yang, Zhongjiang Yan(✉), and Bo Li

School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China
2020262260@mail.nwpu.edu.cn, {yangmao,zhjyan,libo.npu}@nwpu.edu.cn

Abstract. With the rapid development of wireless networks, wireless local area networks (WLAN) are becoming more and more complex and densely deployed, resulting in a significant increase in the time consumption of traditional serial simulations. Aiming at the time consumption problem of traditional discrete-event-based WLAN serial simulation, A parallel simulation method is proposed based on offline learning with non-uniform time slices, which effectively reduces the time consumption. Firstly, the parallel simulation task is modeled as a problem of completing the simulation task within a given time consumption threshold constraint based on the processes pool. Secondly, the time consumption factor is obtained by offline learning of the simulation platform. Thirdly, the parallel simulation algorithm of non-uniform time slice division (NUTSD) based on the time consumption factor is proposed to analyze and solve the problem. Finally, the method is simulated and verified. The simulation results show that this method can greatly reduce time consumption.

Keywords: WLAN · Processes Pool · Offline Learning · Non-Uniform Time Slice · Parallel Simulation

1 Introduction

With the development of WLAN [1], the research on WLAN is deepening, and simulation is one of the main ways to study WLAN [2]. In the study of IEEE 802.11ax [3], due to the diverse services of simulation, the number of nodes is increasing and the deployment is becoming more and more intensive [4], resulting in the traditional serial simulation taking a too long time to meet the needs of simulation. So, it is urgent and necessary to develop a parallel simulation for IEEE 802.11ax.

Parallel simulation is a more effective simulation method compared with serial simulation [5], which includes space parallelism and time parallelism. The space parallel method divides the space into a group of subspaces, such as dividing several nodes into several groups, and each logical process is responsible for

D.-J. Deng et al. (Eds.): SGIoT 2022, LNICST 497, pp. 161–171, 2023.
https://doi.org/10.1007/978-3-031-31275-5_17

completing the tasks of each subspace in the entire simulation time. The time parallel method divides the entire simulation time into a set of consecutive segments, and each logical process is responsible for completing the task in one of the segments. At present, there have been many studies on parallel simulation. For wireless network simulation, the space parallel method is mainly based on NS2 [6] (Network Simulator version 2) and NS3 [7] (Network Simulator version 3), which used MPI [8] (Message Passing Interface) to realize the communication between logical processes. Yuan Jin [9] used NS3 and proposed an algorithm to divide a core network into parts and each part of the network is simulated by one process, which reduced simulation time consumption. Wang Lei [10] designed a parallel network simulation system based on NS2, each function module of the simulation system is simulated on different processes, and MPI is used to provide mutual communication between processes to realize parallel simulation. Wu Qian [11] proposed a method that used MPI to realize multi-machine parallelism based on NS2, the task is assigned to different parallel processes by a parallel scheduler, and the message transmission between parallel processes is completed by the MPI group communication interface, the results show that the simulation acceleration effect is obvious in large scale simulation. Huang Yu et.al [12] completed the parallel detection in the telecommunication network based on MPI for a large-scale backbone communication network, each functional module was implemented on different processes, and the detection rate has improved to some extent. Time parallel simulation method, most of the existing work is to study how to couple time slices after dividing the simulation time. Richard M et. al[5] mentioned that after dividing the time, because the starting state of each process may depend on the state of the simulation ending in the previous period, time slices couple has a great influence on the accuracy of simulation results. However, the above studies have not discussed the time parallel method for high-density wireless networks such as 802.11ax. Since the number of nodes in 802.11ax is too large and interaction between nodes is frequent, performing network simulation using the spatial parallelism method is becoming complex.

We propose a time parallel simulation method with non-uniform time slices based on offline learning to solve the parallel simulation problem of 802.11ax. The simulation verification shows that this method can greatly reduce the simulation time consumption under the premise of guaranteeing the consistency of simulation results with traditional simulation methods.

2 System Model

2.1 Simulation Model Based on Process Pool and Non-uniform Time Slice

A process pool consists of resource processes and management processes. The management process controls communication between internal resource pools and external ports, tasks for controlling internal resource pools include task allocation and process pool status management, and tasks for external communication ports include receiving tasks and status feedback. Resource processes are responsible for internal interprocess communication and task execution, and internal interprocess communication includes task acceptance and status feedback.

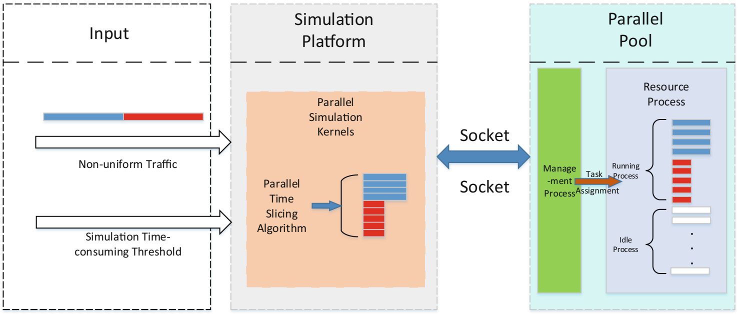


Fig. 1. Simulation model based on process pool and non-uniform time slice.

Figure 1 is a parallel simulation model based on process pools and non-uniform time slices. The inputs of the whole simulation model include non-uniform traffic and simulation time consumption threshold. The process of parallel simulation is the simulation platform divides the simulation time through the non-uniform time slice division algorithm, and then sends the simulation task to the process pool through the socket. The process pool receives the task execution and returns the result to the simulation platform through the socket.

2.2 Problem Description

WLAN simulation tasks can be described as that complete simulation tasks within a given simulation time consumption threshold when the input traffic is non-uniform. It takes a long time for serial simulation to complete the whole simulation, which obviously can't complete the simulation task within the threshold. Therefore, we build a mathematical model based on parallel simulation to solve this problem. The non-uniform traffic is divided into multiple time slices by the NUTSD algorithm. The simulation task of each time slice is completed independently by a process, and the final results are obtained by summarizing the results of all processes.

The mathematical model is as follows, m, T, and $T_{threshold}$ represent the number of idle processes in the process pool, the simulation time consumption, and the time consumption threshold, respectively. $TR_1, TR_2, \cdots, TR_n$ represent the traffic rate, and t_1, t_2, $\cdots$, t_n represent the simulation time of each traffic rate, where n is the number of traffic rate. The time consumption factor is shown in Eq. (1), indicating the relationship between simulation time and simulation time consumption for different traffic rates, where f is a function, and u is the time consumption factor.

$$u_{\mathrm{i}} = f\left(\mathrm{TR}_i \mathrm{Net}\right) \begin{cases} \mathrm{TR}_i \in (1e3, 1e4, \cdots, 1e9) \\ Net \in (CBR, Possion) \\ i \in (1, 2, 3, \cdots, n) \end{cases} \tag{1}$$

When using serial simulation to perform simulation tasks, the simulation time consumption will be far greater than the simulation time consumption threshold, as shown in Eq. (2).

$$T >> T_{threshold} \tag{2}$$

Obviously, serial simulation can't solve the problem. According to the NUTSD algorithm, the simulation time is divided into several time slices, and each time slice is assigned to different processes for simulation.

$$T < T_{threshold} \tag{3}$$

Equation (3) is satisfied if Eq. (4) is satisfied, where T_{proi} is the simulation time consumption for each process. Therefore, when the longest simulation time consumption in all processes satisfies Eq. (3), the simulation task can be completed.

$$T = \max\left(T_{pro1}, T_{pro2}, \cdots, T_{pron}\right), T_{proi}\left(i \in (1, 2, \cdots n)\right) \tag{4}$$

The estimation of simulation time consumption based on the time consumption factor is shown in Eq. (5), and the number of required processes P is shown in Eq. (6), where $\lceil\ \rceil$ is rounding up the result. The simulation time consumption T_{proi} for each process is shown in Eq. (7).

$$T = t_1 \cdot u_1 + t_2 \cdot u_2 + \cdots + t_n \cdot u_n = \sum_{i=1}^{n} t_i \cdot u_i \tag{5}$$

$$P = \left\lceil \frac{T}{T_{threshold}} \right\rceil \tag{6}$$

$$T_{proi} = \frac{T}{P} \tag{7}$$

Eventually, the simulation duration of each process can be obtained according to Eqs. (1) and (7), and Eq. (3) can be satisfied when the number of execution processes is P.

3 Offline Learning

In this section, we will discuss the time consumption factor, which was mentioned earlier in Sect. 2, and will show the steps to obtain the time consumption factor through offline learning. A data packet will go through several states in the simulation process, and each state generates several events. Moreover, the execution complexity of each event is different, and the time consumption is also different.

Table 1. Package Status and Example of Major Events.

Status	Major Events
Application Layer Generation Package	*IntraApplication*
	StartApplication
Successfully Sent the Package	*SendDataProcess*
Successfully Received the Packet	*SendAck SendCTS*
Retransmit Successful Packets After Receiving Failure	*HandleCTSTimeOut*
	HandleCTSTimeOut
Receive Failure to Retransmit Timeout Packet	*Lowest Level Heading.*
The Remaining Packets in the Queue After the End	*Lowest Level Heading.*

During the IEEE 802.11ax simulation, the data packets are divided into several states and major events as shown in Table 1. When there have enough events, we can assume that the number of events follows the law of large numbers, and we can calculate the average number of events experienced by each data packet from generation to destruction.

Table 2. The Number of Events Corresponding to the Data Packet

	Number of Packages	Number of Events	Total Events
Application Layer Generation Package	X_1	C_1	X_1C_1
Successfully Sent the Package	X_2	C_2	X_2C_2
Successfully Received the Package	X_3	C_3	X_3C_3
Retransmit Successful Packets After Receiving Failuret	X_4	C_4	X_4C_4
Receive Failure to Retransmit Timeout Packet	X_5	C_5	X_5C_5
The Remaining Packets in the Queue After the End	X_6	C_6	X_6C_6

The offline learning can be represented by a mathematical model. t, T, y and X represent the simulation time, the simulation time consumption, the number of events corresponding to t, and the number of data packets corresponding to t, respectively, others are shown in Table 2.

In the IEEE 802.11ax simulation system, when the simulation time is t, X is shown in Eq. (8), and y is shown in Eq. (9).

$$\begin{aligned} X &= X_1 + X_2 + X_3 + X_4 + X_5 + X_6 \\ &= f_1(t) + f_2(t) + f_3(t) + f_4(t) + f_5(t) + f_6(t) \end{aligned} \tag{8}$$

$$y = C_1X_1 + C_2X_2 + C_3X_3 + C_4X_4 + C_5X_5 + C_6X_6 = g(f(t)) \tag{9}$$

$$T = h(y) = h(g(f(t))) \tag{10}$$

T can be calculated by Eq. (10), and the relationship between T and t is very complicated. Therefore, in this paper, a large amount of data is obtained by simulating the 802.11ax simulation system, and the data is fitted to obtain the relationship between T and t, that is, the time consumption factor, denoted by u.

Table 3. Simulation Parameter Table

Simulation Parameters	Parameters
Traffic Type-Topology	*CBR-DL-1AP1STA*
	CBR-DL-6AP3STA
	Possion-DL-1AP3STA
Simulation Time	*10*s
Traffic Rate	*1e8(5 s)–1e3(5 s)*
Packet Size	*1500bits*
Bandwidth	*20M*
MCS Value	*9*

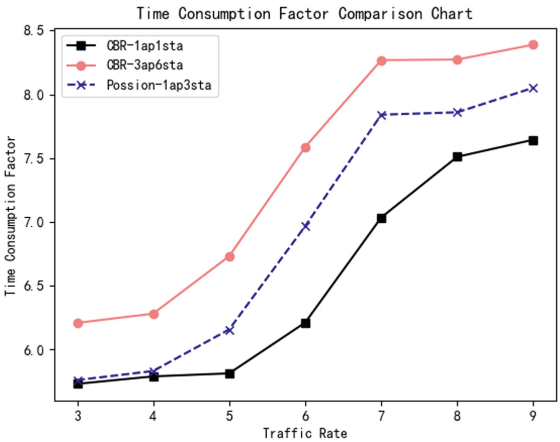


Fig. 2. Different Traffic-Topology Time Consumption Factor Result

A large number of results were obtained through simulation under the simulation parameters shown in Table 3, and the time consumption factor was obtained by fitting the results. Figure 2 shows the time consumption factor results for different traffic rates and different topologies.

4 Non-uniform Time Slice Parallel Simulation Algorithm Based on Offline Learning

Based on Sect. 3, this section proposes a non-uniform time slice parallel simulation algorithm based on offline learning. The inputs of the algorithm include the rate of each traffic, the simulation time of each traffic rate, the time consumption factor, the simulation time consumption threshold, and the total number of processes in the process pool. The output is the simulation time consumption. Firstly, the estimated simulation time of all services is calculated according to the time consumption factor. Then the number of processes required to complete the simulation task within the simulation time consumption threshold and the corresponding simulation time consumption of each process are calculated. Finally, the simulation time required for each process is calculated in reverse. The algorithm can not only complete the simulation task with the least process within the simulation time consumption threshold but also ensures the load balance of the process.

The pseudo-code of the algorithm is as follows:

Algorithm 1: NUTSD Algorithm

---

input : Tacffic Rate TR_i and Simulation Time of Each Rate t_i and Time consumption Factor u_i and Simulation Time consumption Threshold $T_{threshold}$ and Total Number of Processes P_{total}

output: Simulation Time consumption T

Begin

foreach TR_i **do** T_{total}+=$u_i \cdot t_i$;

Calculation of Simulation Time Consumption for All Traffic

$\boldsymbol{P_{need}} = \left\lceil \frac{T_{total}}{T_{threshold}} \right\rceil$

Calculate the number of processes required to complete the simulation within the simulation time consumption threshold

$\boldsymbol{P_{time} = T_{total}/P_{need}}$

Calculating simulation time consumption for each process

foreach $i < P_{need}$ **do** ***CalculateSimulationtime***();

Calculate the simulation time for each process

if $P_{need} <= P_{total}$ **then** ***StartSimulation***();

Start Simulation

else Print("Insufficient idle processes");

End

---

5 Simulation

5.1 Simulation Environment and Parameter Settings

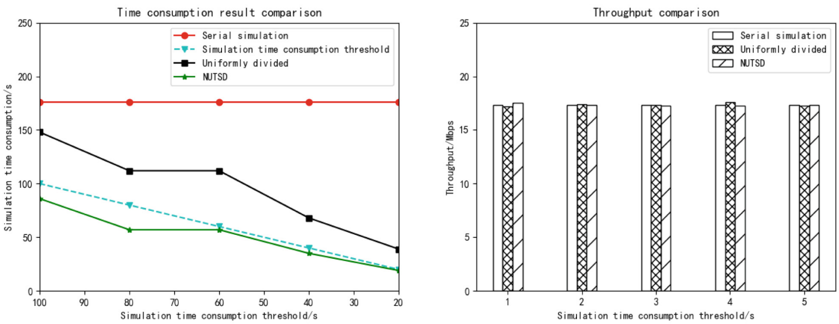
This paper is implemented and verified on the IEEE 802.11ax simulation platform [13] of our laboratory. The main simulation verification parameters are shown in Table 4.

Table 4. Simulation Parameter Table

Simulation Parameters	Parameters
Traffic Type-Topology	*CBR-DL-1AP1STA*
	CBR-DL-6AP3STA
	Possion-DL-1AP3STA
Traffic Rate	*1e8(5s)–1e3(5s)*
Packet Size	*1500bits*
Bandwidth	*20M*
MCS Value	*9*
OFDMA	*Open*

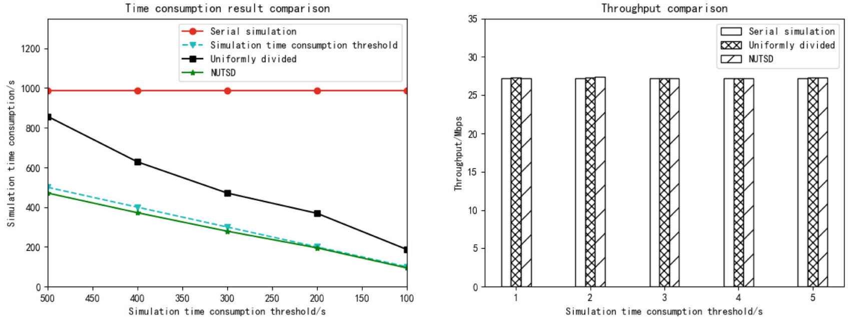
5.2 Simulation Results and Analysis

Figure 3(a) shows the results of the serial simulation, uniform divided and non-uniform divided simulation time consumption changes with preset simulation time consumption thresholds in the downlink simulation verification environment of CBR traffic under 1AP1STA topology. Figure 3(b) is the serial simulation, uniformly divided and non-uniformly divided throughputs vary with the preset simulation time consumption threshold, and the packet loss rate is consistent with serial simulation.



(a) CBR-1ap1sta Simulation Time Consumption Comparison Result (b) CBR-1ap1sta Throughput Comparison Result

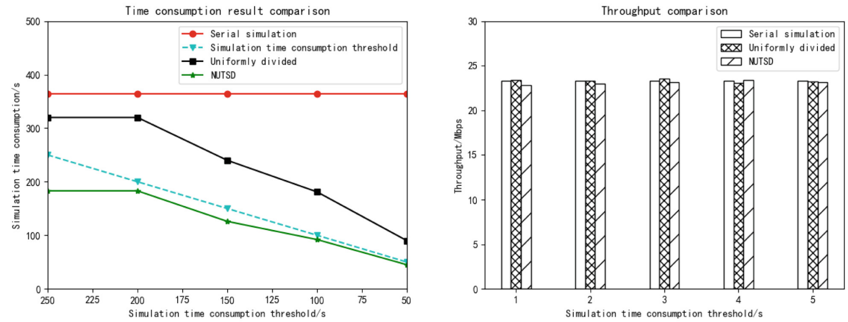
Fig. 3. CBR-1ap1sta Comparison Result



(a) CBR-3ap6sta Simulation Time Consumption Comparison Result (b) CBR-3ap6sta Throughput Comparison Result

Fig. 4. CBR-3ap6sta Comparison Result

Figure 4(a) shows the results of the serial simulation, uniform division, and non-uniform division simulation time consumption changes with the preset simulation time consumption threshold in the downlink simulation verification environment of the CBR traffic under the 3AP6STA topology. Figure 4(b) shows the serial simulation, uniformly divided and non-uniformly divided throughputs vary with the preset simulation time consumption threshold, and the packet loss rate is consistent with serial simulation.



(a) Possion-1ap3sta Simulation Time Consumption Comparison Result (b) Possion-1ap3sta Throughput Comparison Result

Fig. 5. Possion-1ap3sta Comparison Result

Figure 5(a) shows the results of the serial simulation, uniform division, and non-uniform division simulation time consumption changes with the preset simulation time consumption threshold in the downlink simulation verification environment of the poisson traffic under the 1AP3STA topology. Figure 5(b) shows the serial simulation, uniformly divided and non-uniformly divided throughputs vary with the preset simulation time consumption threshold, and the packet loss rate is consistent with serial simulation.

From the above three sets of simulation results, it can be seen that for different traffic rates and different topologies, the simulation task which was proposed in Sect. 2.2 can be solved within the time consumption threshold through the time consumption factor and NUTSD algorithm. The throughput and packet loss rate are consistent with serial simulation. Compared with serial simulation and uniform time slice parallel simulation, non-uniform time slice parallel simulation can greatly reduce the simulation time consumption.

6 Conclusion

To solve the time consumption problem of traditional serial simulation in WLAN, this paper proposes a parallel simulation method based on offline learning with non-uniform time slice division. Through the modeling and analysis of the problem and the implementation of the NUTSD algorithm, the simulation verification is carried out on the IEEE 802.11ax simulation platform. The results show that the method can effectively reduce the simulation time consumption, and has a strong application prospect and value in WLAN simulation.

Acknowledgement. This work was supported in part by the National Natural Science Foundations of CHINA (Grant No. 61771392, No. 61871322 and No. 61771390), and Science and Technology on Avionics Integration Laboratory and the Aeronautical Science Foundation of China (Grant No. 201955053002 and No. 20185553035).

References

1. Hao, L., Zehua, G., Feng, G., Ronghua, Z.: IEEE 802.11 WLAN standard research. Comput. Appl. Res. **26**(05), 1616–1620 (2009)
2. Chunguang, M., Jiansheng, Y.: NS-3 Network Simulator. Name Posts and Telecommunications Press, Beijing (2014)
3. Qu, Q., Li, B., Yang, M.: Survey and performance evaluation of the upcoming next generation WLANs standard-IEEE 802.11ax. Mob. Netw. Appl. **24**(03), 1461–1474 (2019)
4. IEEE802.11ax/D5.1.Wireless Medium Access Control(MAC) and Physical Layer(PHY) Specifications: Enhancements for High Efficiency in Frequency Bands between 1 GHz and 6 GHz. (2019)
5. Fujimoto, R.M.: Parallel and Distributed Simulation Systems. Canada (2000)
6. Xioayan, W., MIngchun, Z.: Research and Application of Network Simulation Based on NS2. Comput. Simul. (2004)
7. Xinyu, R., Yuan, L.: Analysis and exploration of network simulation NS3. Comput. Technol. Dev. (2018)
8. Zhihong, Z.: Parallel computing based on MPI. China University of Geosciences, Beijing (2006)
9. Jin, Y.: Research on network parallel simulation method based on NS3. University of Electronic Science and Technology (2017)

10. Lei, W.: Research and implementation of parallel network simulation task optimization method. The Harbin University of Technology (2006)
11. Qian, W.: Design and implementation of NS2 multi-machine parallel simulation scheduler. Nanjing University of Posts and Telecommunications (2016)
12. Yu, H., Guangmin, H.: Development of computer parallel simulation platform for backbone communication network. Microcomput. Inf. **26**(16) (2010)
13. Yang, M., Li, B., Yan, Z.: MAC technology of IEEE 802.11ax: progress and tutorial. Mob. Netw Appl. **26**, 1122–1136 (2021)