Performance Evaluation of Modified Tree Algorithm for Supporting Traffic with Different Priority Requirements

Warakorn Srichavengsup[1] Kanticha Kittipeerachon[2*]

[1,2*]*Computer Engineering Robotics and Technology Research Laboratory : CERT,*
Faculty of Engineering, Thai-Nichi Institute of Technology, Bangkok, Thailand

*Corresponding Author. E-mail address: kanticha@tni.ac.th

Received: 13 October 2022; Revised: 21 November 2022; Accepted: 24 November 2022
Published online: 22 December 2022

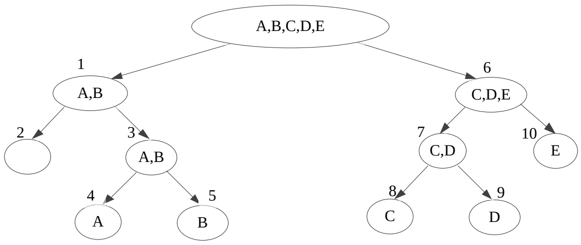
Abstract

In this paper, we are introducing 3 new algorithms that take high-priority services into account for more delay-sensitive multimedia traffic, namely, CFS, DAP type 1 and DAP type 2 algorithms, an extension of Binary tree algorithm. The CFS algorithm can be derived from Binary tree algorithm by setting the access probability of the first slot for class 1 users equal to 1. DAP type 1 algorithm can be obtained from Binary tree algorithm by adjusting the access probability of the first slot for class 1 users to the desired value. DAP type 2 algorithm can be derived from either Binary tree and DAP type 1 algorithm by adjusting the access probability of the first slot for class 1 users and the access probability of the first slot for class 2 users to the desired values. The goal is to make each type of traffic compliant with QoS requirements. The simulation results show that DAP type 2 algorithm has more feasible QoS indexes than the CFS and DAP type 1 algorithms. So it is more efficient and flexible to adjust system parameters to meet QoS requirements than CFS and DAP type 1 algorithm. In addition, we can conclude that the access probability of the first slot for class 1 users and class 2 users are the important parameters that must be set properly. So that the system can control QoS and still maintain high delay performance.

Keywords: Binary tree algorithm, Priority, QoS

I. INTRODUCTION

The most popular communication today is high-speed wireless communication, which can support a large number of users. If a large number of users would like to send data at the same time, it will cause a data collision and no user will be successful in transmitting data. Therefore, many Medium Access Control (MAC) protocols were designed. MAC protocols can be classified into contention free [1]–[3] and contention based [4], [5] protocols. Contention free protocols allocate resources to each user, ensuring no data collisions occur. Examples of this type of protocol are TDMA [6], [7], where time is allocated for each user to transmit data, and FDMA [8], [9], where each user is assigned a different frequency to transmit data. Using different times or frequencies will definitely not cause a collision. But these protocols have a disadvantage. If some users do not have the data to send. This will make the time or frequency that is reserved cannot be used by others. In order to solve this problem, the contention based algorithms were introduced. For contention based algorithms, if any user would like to transmit data, they will compete for access to the channel. The examples of this type of algorithm are ALOHA protocol [10], [11], slotted ALOHA protocol [12], [13], Carrier Sense Multiple Access (CSMA) protocol [14], [15] and binary tree algorithm [16], [17]. Binary tree algorithm has the advantage of being able to support a large number of users by splitting the collision-related users into 2 subgroups. When another collision occurs, the users involved in the collision are divided into two subgroups. This process continues until all users successfully access the channel. Fig. 1 illustrates the example of the collision resolution mechanism of Binary tree algorithm. From the figure, users A, B, C, D and E have a delay time of 4, 5, 8, 9 and 10, respectively.



1	2	3	4	5	6	7	8	9	10
A,B		A,B	A	B	C,D,E	C,D	C	D	E

Figure 1: Collision resolution mechanism of Binary tree algorithm

Nowadays, in the era of high-speed communication, there are high demands on data transmission that require different quality of service. Although the binary tree algorithm can support a large number of users, however, it cannot be applied to systems where different users require different quality of service, such as audio or video transmission. Therefore, this paper presents 3 new algorithms, namely, CFS, DAP type 1 and DAP type 2 algorithms to support traffic with different priority requirements.

The paper is structured as follows. In section II, we shall describe the details of CFS, DAP type 1 and DAP type 2 algorithms. Results and discussions will be presented in section III. Finally, conclusions are presented in Section IV.

II. PROPOSED ALGORITHMS

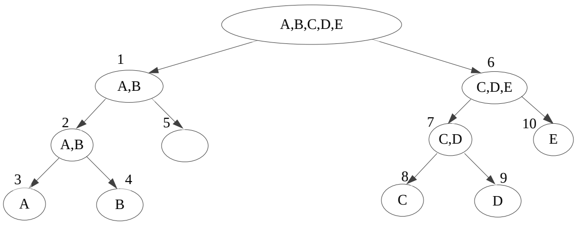
This paper presents 3 algorithms that can support traffic with different priority requirements, which are CFS, DAP type 1 and DAP type 2. Each algorithm divides users into two groups, which are class 1 and class 2 users. Let class 1 users have the higher priority than class 2 users, i.e. class 1 users need less time delay than class 2 users. Moreover, we introduce a QoS index parameter to compare the quality of service of class 1 and class 2 users. The QoS index is defined as the ratio between the average delay time of class 2 users and the average delay time of class 1 users.

The details of all proposed algorithms are as follows.

A. Choosing First Slot (CFS)

In this algorithm, class 1 users always accesses the first slot. As illustrated in Fig. 2, it can be seen that user A, a class 1 user, accesses the first channel in order to gain access to the channel before other users. Whereas class 2 users randomly select one of the 2 slots, i.e. class 1 user accesses slot 1 with probability 1, while class 2 users randomly choose slots 1 and 2 with probability equal to 0.5.

For CFS algorithm, only one user can be a class 1 user. This is because if there are more than one user, there will be endless collisions. Because all class 1 users access to the first slot, causing an inevitable collision.



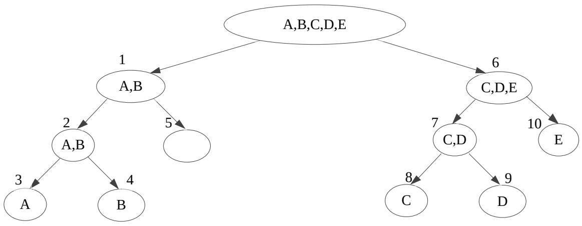
1	2	3	4	5	6	7	8	9	10
A,B	A,B	A	B		C,D,E	C,D	C	D	E

Figure 2: Collision resolution mechanism of CFS algorithm

B. Different Access Probability Type 1 (DAP Type 1)

For this algorithm, the probability of accessing the first slot of class 1 users is greater than 0.5. Therefore, class 1 users have the greater chance of accessing the first slot than the class 2 users, as illustrated in Fig. 3. In the figure, the access probabilities of class 1 and class 2 users are 0.7 and 0.5 respectively. Class 2 users behave like normal users in Binary tree algorithm. Let users A, B and C are class 1 users and users D and E are class 2 users. User A, a class 1 user, randomly selects a number between 0 and 1. If the random number is less than 0.7. User A will choose to access the first slot. But if the random number is greater than or equal to 0.7. User A will access the second slot. In this example, the random number of user A is equal to 0.65, so User A chooses to access the first slot. While User D, a class 2 user, randomly selects a number between 0 and 1. If the random number is less than 0.5. User D will choose to access the first slot. In this example, the random number of user D is equal to 0.9, so User B chooses to access the second slot. This process is repeated until all users have successfully accessed the channel.

Class	1	1	1	2	2
User	A	B	C	D	E
P	0.7	0.7	0.7	0.5	0.5
rand	0.65	0.28	0.84	0.9	0.78
rand	0.42	0.51	0.55	0.36	0.85
rand	0.68	0.82	0.24	0.65	



1	2	3	4	5	6	7	8	9	10
A,B	A,B	A	B		C,D,E	C,D	C	D	E

Figure 3: Collision resolution mechanism of DAP type 1 algorithm

C. Different Access Probability Type 2 (DAP Type 2)

For this algorithm, class 1 users have the higher chance of accessing the first slot than the second slot, as is the case with DAP type 1 algorithm. However, the access probability of class 2 users to access the first slot is not necessarily 0.5, as shown in Fig. 4. In the figure, the access probability of class 1 users is 0.7, while the access probability of class 2 users is 0.3. User A, a class 1 user, has a chance to access the first and second slots of 0.7 and 0.3 respectively. That is class 1 users have the higher chances of accessing the first slot than the second slot. Whereas User D, a class 2 user, has a probability of accessing the first slot of 0.3 and a probability of entering the second slot of 0.7. This means that class 2 users have the higher chances of accessing the second slot than the first slot. Therefore, in this case the average delay time of class 1 users will be less than the average delay time of class 2 users.

Journal of Engineering and Digital Technology (JEDT)
Vol.10 No.2 July - December 2022

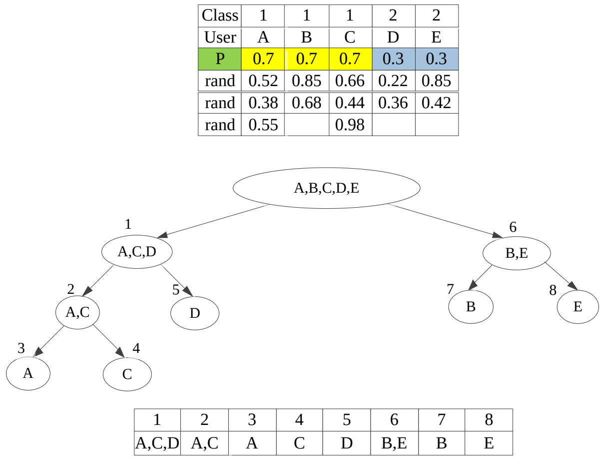


Figure 4: Collision resolution mechanism of DAP type 2 algorithm

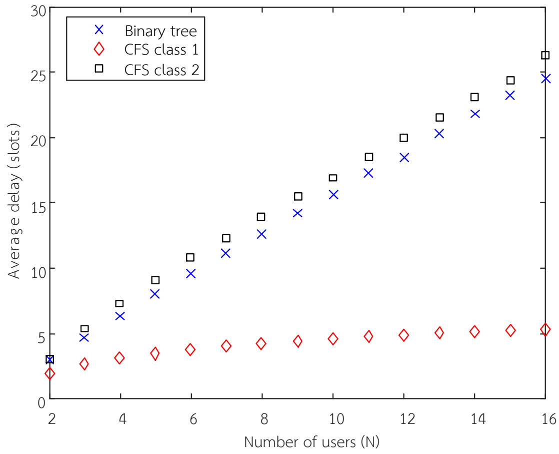


Figure 5: The average delay time vs the number of users for the Binary tree and CFS algorithms

III. RESULTS AND DISCUSSION

In this section, we shall investigate the performance of the proposed algorithms. The details are as follows.

A. Performance of CFS Algorithm

As can be seen in Fig. 5, the average delay time of class 1 and class 2 users increase with the number of users in the system. Since the number of slots is equal to 2, when there is a large number of users in the system, there will be frequent collisions. We can notice that the average delay time of the class 1 user is slightly increased. This is because class 1 user always accesses the first slot. If a collision occurs, the class 1 user and class 2 users involved in the collision are divided into subgroups to resolve the collision. This group of users will succeed in accessing the channel before other group of users. In addition, we found that the average delay time of class 2 users is higher than the average delay time of normal user for Binary tree algorithm. This is because a class 1 user always accesses the first slot and this increases the chance of a collision between class 1 user and some of class 2 users. The multiple slots is required to solve the collision. As a result, the average delay time of class 2 users is high.

Fig. 6 shows the relationship between the QoS index value and the number of users in the system. This figure shows that the QoS index increases with the number of users. This is because the QoS index is equal to the ratio between the average delay time of class 2 and class 1 users. From Fig. 5, it can be seen that the average delay time of class 2 users has a much higher rate of increase than the average delay time of class 1 users, resulting in the ratio increasing with the number of users.

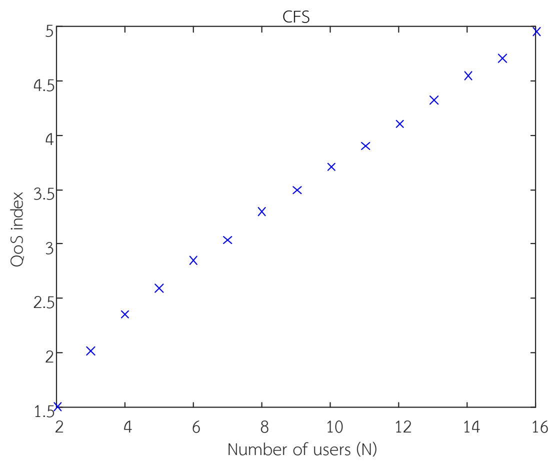


Figure 6: The Qos index vs the number of users for the CFS algorithm

B. Performance of DAP Type 1 Algorithm

Fig. 7 shows a comparison of the average delay time between class 1 and class 2 users of the DAP type 1 algorithm. The figure shows only the case that the access probability of class 1 users greater than or equal to 0.5. This is because in this paper we assign class 1 users a higher priority than class 2. users. If class 1 users use the access probability less than 0.5, then class 1 users will have a higher average delay time than class 2 users.

From the figure, we found that when increasing the probability of accessing the first slot of class 1 users the average delay time of class 1 users tends to decrease. However, when increasing the probability of accessing the first slot of class 1 users too much, the average delay time of class 1 users will be much higher. This is because using proper access probabilities will give class 1 users an appropriate channel access rate. This causes infrequent collisions between users. However, when increasing the access probability too much, tmhe chance of collisions between class 1 users is high, resulting in a noticeable increase in the average delay time of class 1 users.

When considering the average delay time of class 2 users, it can be seen that the average delay time of class 2 users increases with the access probability of class 1 users because increasing the access probability of class 1 users would increase the chance of collisions between class 1 and class 2 users. In addition, when the number of class 1 users increases, increasing the access probability of class 1 users will increase the average delay time of class 2 users. Because there is a high chance of collision between class 1 users. Therefore, most class 2 users have to wait for collision-related class 1 users to successfully access the channel.

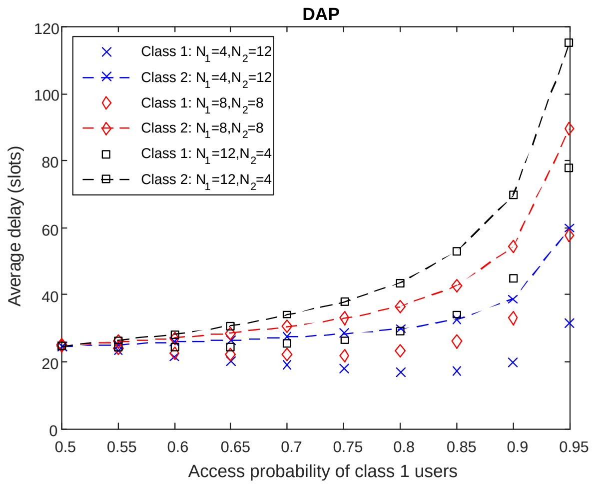


Figure 7: The average delay time vs the access probability of class 1 for the DAP type 1 algorithm when varying the number of users in each class

Fig. 8 shows the relationship between the QoS index and the access probability of class 1 users. It can be seen that when the access probability of class 1 users is 0.5, the QoS Index is 1. This is because all users use a probability value of 0.5, the users of both classes have the same chance of success. In this specific condition, the DAP type 1 algorithm becomes Binary tree algorithm.

When increasing the access probability of class 1 users, the QoS index tends to increase. This is because the average delay time of class 2 users increases with the access probability of class 1 users while the average delay time of class 1 users tends to slightly increased. The QoS Index is maximally increased at a certain access probability of class 1 users. Increasing more access probability of class 1 users will decrease the QoS index. This is because increasing the access probability of class 1 too much will increase the chance of collisions between class 1 users and multiple slots are required to solve the collision problem.

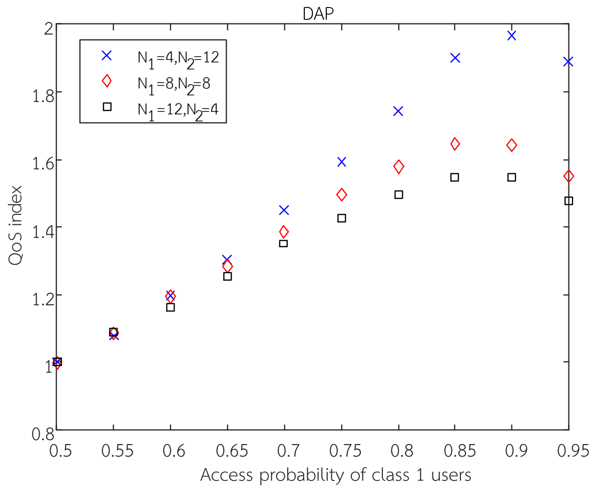


Figure 8: The QoS index vs the access probability of class 1 users for the DAP type 2 algorithm when varying the number of users in each class.

C. Performance of DAP Type 2 Algorithm

For conveniences, these notations are used in the following discussions:

p_1 = the access probability of the first slot for class 1 users.

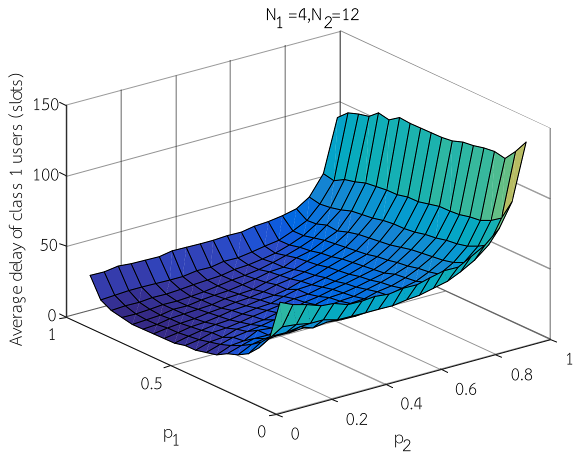
p_2 = the access probability of the first slot for class 2 users.

N_1 = the number of class 1 users.

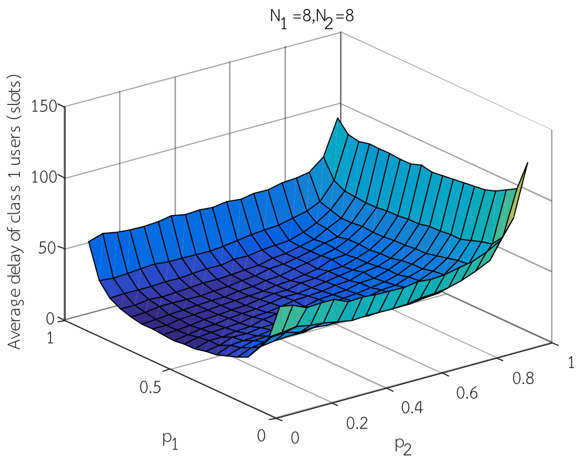
N_2 = the number of class 2 users.

Comparing Figs. 9(a), 9(b) and 9(c), it is found that if the number of class 1 users is greater than the number of class 2 users, changing the p_1 value will cause the average delay time of class 1 users to change rapidly. While changing the p_2 value will cause the average delay time of class 1 users to change slowly. On the contrary, if the number of class 2 users is greater than the number of class 1 users, changing the p_2 value will cause the average delay time of class 1 users to change rapidly. While changing the value of p_1 will cause the average delay time of class 1 users to change slowly.

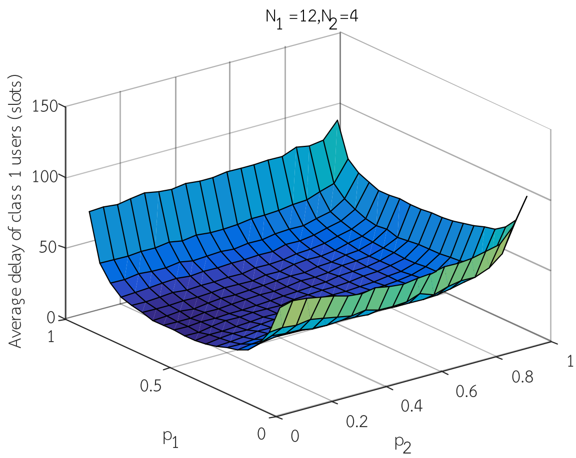
The average delay time of class 1 users is lowest when p_1 is in the range 0.3–0.7 and p_2 is small. This is because using the proper value of p_1 can help reduce the chance of collisions between class 1 users, and when p_2 is small, most class 2 users will access the second slot. This gives class 1 users a chance to achieve faster channel access.



(a)



(b)



(c)

Figure 9: The average delay of class 1 users for the DAP type 2 algorithm as a function of p_1 and p_2 : (a) N_1=4, N_2=12 (b) N_1=8, N_2=8 (c) N_1=12, N_2=4

Fig. 10 shows the relationship between the average delay time of class 2 users and the values of p_1 and p_2. From the figure, the average delay time of class 2 users tends to increase with the value of p_1. This is because when p_1 is large such as 0.95, class 1 users have a high chance of accessing the firsrt slot. Therefore, most class 2 users have to wait for collision-related class 1 users to successfully access the channel.

The average delay time of class 2 users is lowest when p_2 is in the range of 0.3–0.7 and p_1 is small. This is because using medium value of p_2 can help reduces the chance of collisions between class 2 users, and at the lowest p_1 value, most class 1 users will access the second slot. This gives class 2 users a chance to successfully access early slots.

Fig. 11 shows the relationship between the QoS index value with p_1 and p_2. It is found that the QoS index is high when p_2 is low. This is because when p_2 is low, most class 2 users are successful in access the channel at the latter slots. As a result, class 2 users have a high average delay time value, resulting in a higher QoS index. When considering at a small p_2 value, the QoS index increases with p_1 until it reaches a maximum of a certain p_1 value. When p_1 is increased further, the QoS index decreases. This peculiar effect can be explained as the following. When p_1 is too low, most class 1 users will access the second slot, resulting in increased delay of the class 1 user. In the case of high p_1 values, there will be a high probability of collision between class 1 users, resulting in a high delay time and low QoS index value. If we would like to achieve a high QoS index we should use the small p_2 value and use a p_1 value between 0.3–0.7.

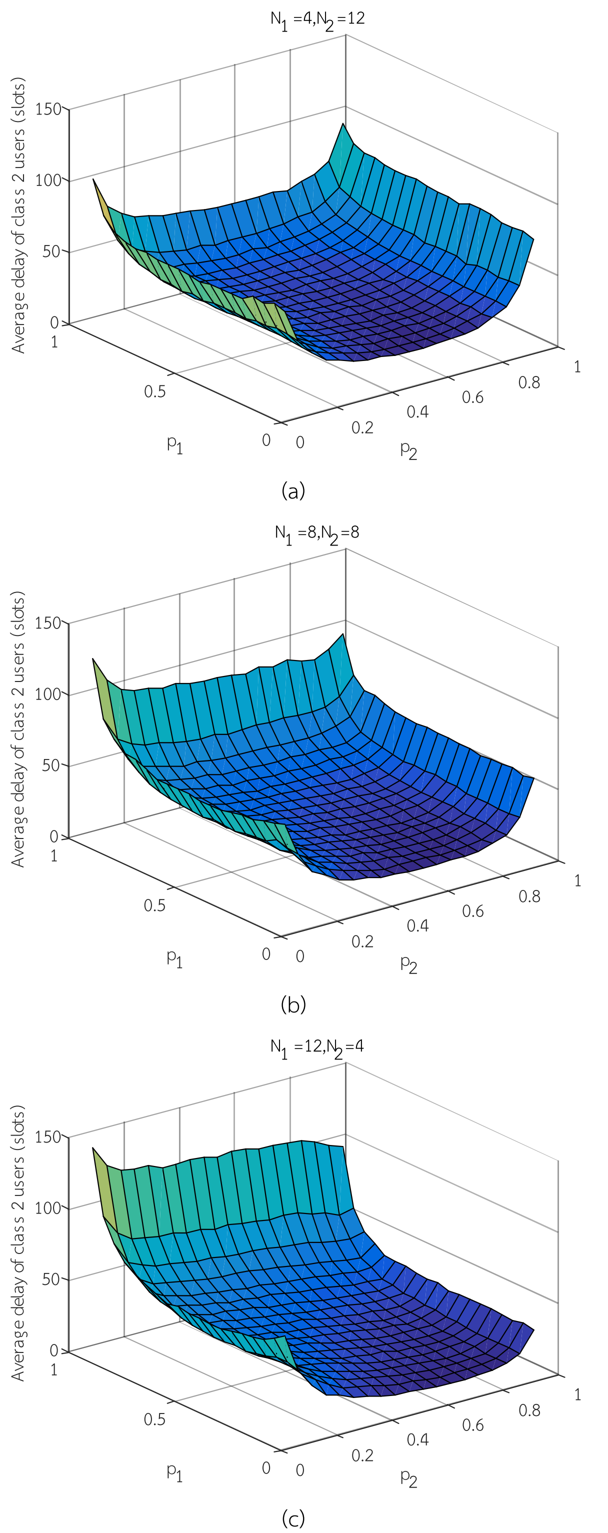


Figure 10: The average delay of class 2 users for the DAP type 2 algorithm as a function of p_1 and p_2 : (a) N_1=4, N_2=12 (b) N_1=8, N_2=8 (c) N_1=12, N_2=4

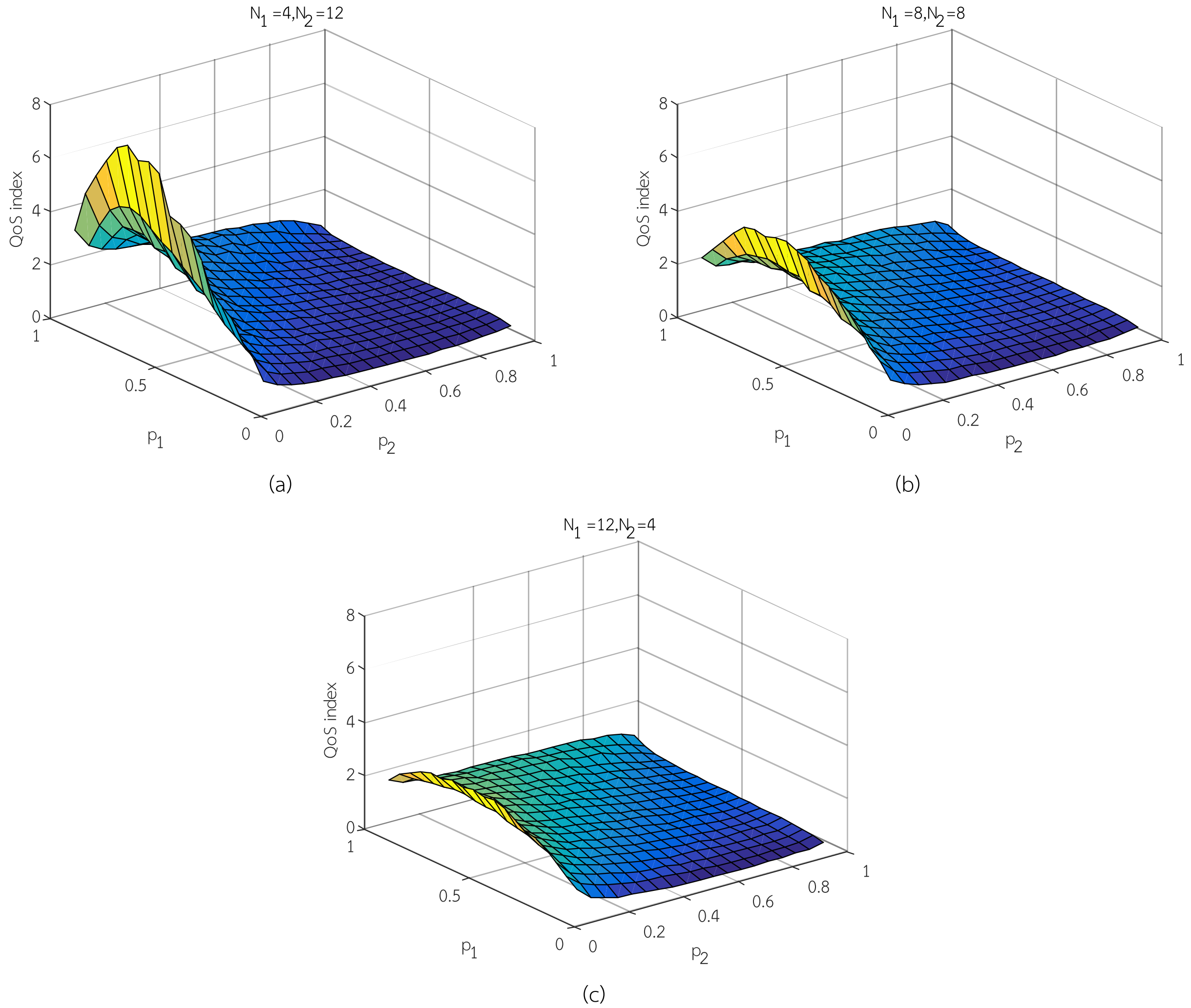


Figure 11: The QoS index for the DAP type 2 algorithm as a function of p_1 and p_2 : (a) N_1=4, N_2=12 (b) N_1=8, N_2=8 (c) N_1=12, N_2=4

Figs. 12 and 13 show the relationship between QoS index value and the average delay time of class 1 and class 2 users respectively. It can be seen that at high QoS index value, the delay time of class 1 user is low. While the delay time of class 2 users is large. We also found that increasing the proportion of class 1 users would decrease the range of QoS index, reducing from 7.6 to 3.15.

When comparing Binary tree, CFS, DAP type 1 and DAP type 2 algorithms as shown in Figs. 14 and 15. We found that the DAP type 2 algorithm has more feasible QoS indexes than the other algorithms. So it is more effective and flexible in adjusting the system parameters to meet QoS requirements than CFS and DAP type 1 algorithms. This is because DAP type 2 algorithm can use both p_1 and p_2 values to adjust the QoS index, whereas DAP type 1 algorithm can use only p_1 value to adjust the QoS index. For the CFS algorithm, p_1 and p_2 are constants of 1 and 0.5, respectively. Moreover, we can observe that the QoS index of Binary tree algorithm is equal to 1. This is because for the Binary tree algorithm all users use the same access probability, which is 0.5.

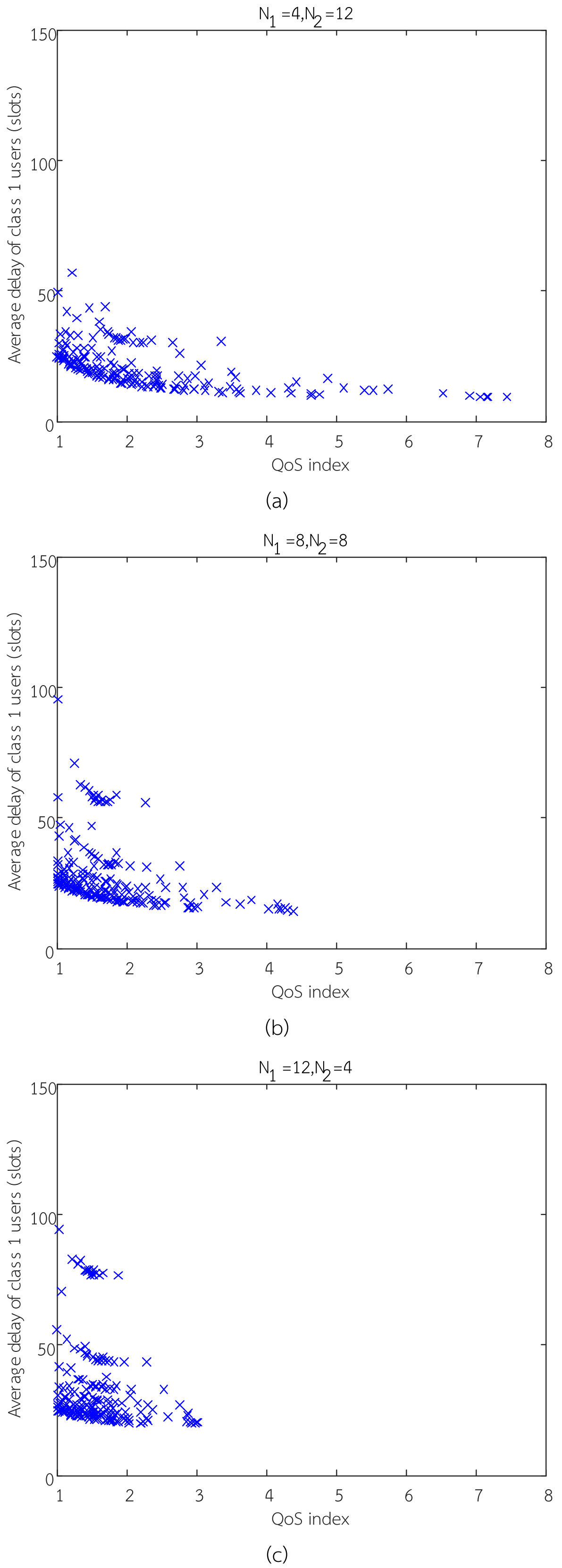


Figure 12: The average delay of class 1 users vs QoS index for the DAP type 2 algorithm : (a) N_1=4, N_2=12 (b) N_1=8, N_2=8 (c) N_1=12, N_2=4

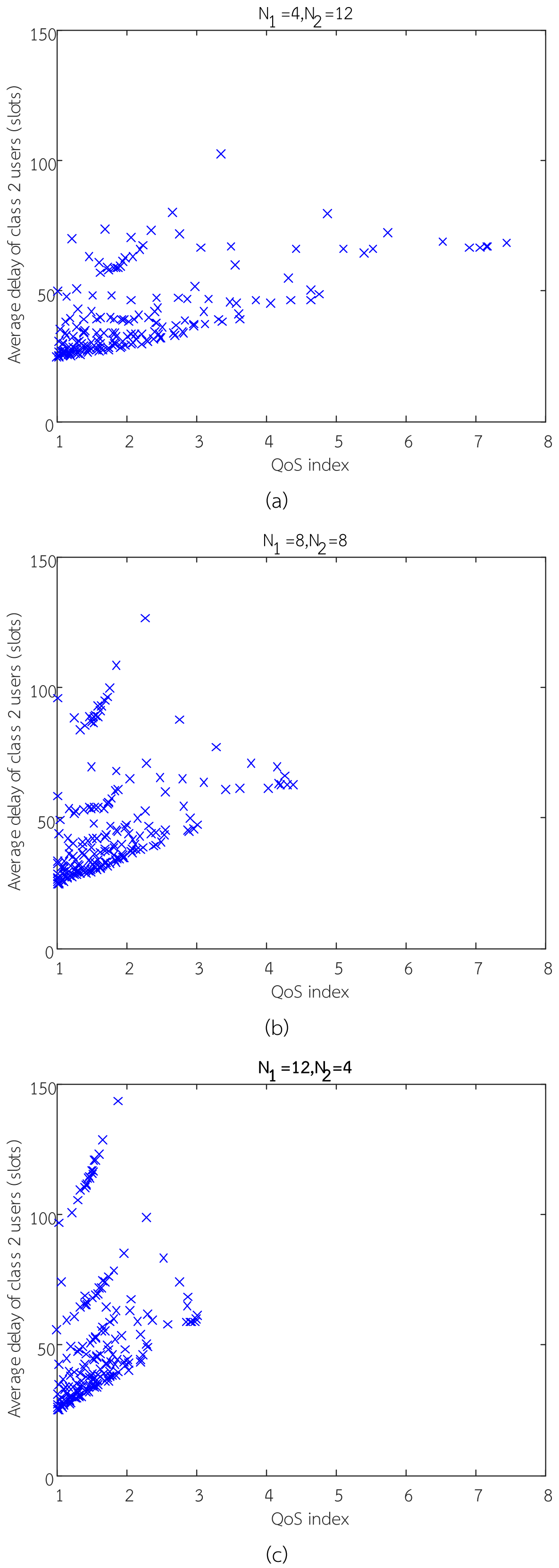


Figure 13: The average delay of class 2 users vs QoS index for the DAP type 2 algorithm : (a) N_1=4, N_2=12 (b) N_1=8, N_2=8 (c) N_1=12, N_2=4

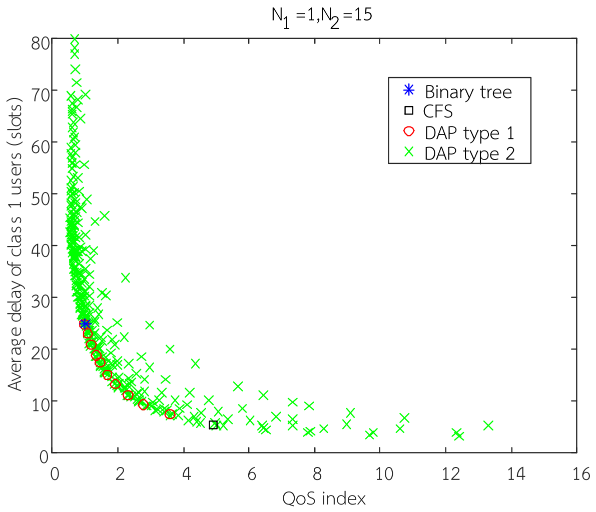


Figure 14: The average delay of class 1 users vs the QoS index for Binary tree, CFS, DAP type 1 and DAP type 2

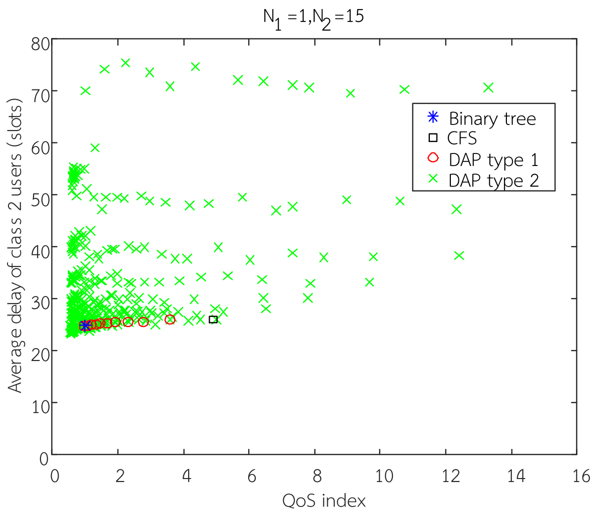


Figure 15: The average delay of class 2 users vs the QoS index for Binary tree, CFS, DAP type 1 and DAP type 2

At the end of the section, we summarize the relations among Binary tree, CFS, DAP type 1 and DAP type 2 algorithms as shown in Fig. 16. It can be seen that all proposed algorithms can be derived from Binary tree algorithm, which can be described as follows. The CFS algorithm can be derived from Binary tree algorithm by setting the value of p_1 equal to 1. DAP type 1 algorithm can be obtained from Binary tree algorithm by adjusting p_1 to the desired value. DAP type 2 algorithm can be derived from either Binary tree and DAP type 1 algorithm by adjusting p_1 and p_2 to the desired values.

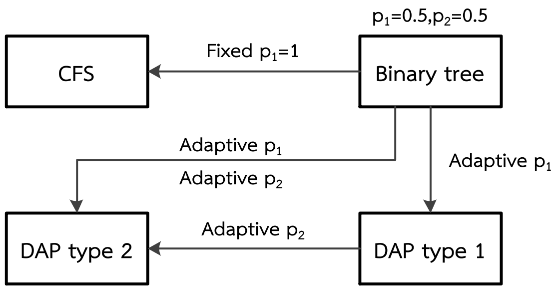


Figure 16: The relations between Binary tree, CFS, DAP type 1 and DAP type 2 algorithms

IV. CONCLUSION

In this paper, we have presented 3 new algorithms, namely, CFS, DAP type 1 and DAP type 2 algorithms to support multi-class traffic with different QoS requirements. All proposed algorithms can be derived from Binary tree algorithm. From the simulation results, we can conclude that the access probability of the first slot for class 1 users and class 2 users are the important parameters that must be set properly. So that the system can control QoS and still maintain high delay performance. When comparing among all proposed schemes in terms of QoS requirements we found that the DAP type 2 algorithm has more feasible QoS indexes than the CFS and DAP type 1 algorithms. So it is more effective and flexible in adjusting the system parameters to meet QoS requirements than CFS and DAP type 1 algorithms.

REFERENCES

[1] G. Auer, S. Videv, B. Ghimire, and H. Haas, “Contention free inter-cellular slot reservation,” *IEEE Commun. Lett.*, vol. 13, no. 5, pp. 318–320, May 2009.

[2] W. H. A. Yuen and W. S. Wong, “A contention-free mobility management scheme based on probabilistic paging,” *IEEE Trans. Veh. Technol.*, vol. 50, no. 1, pp. 48–58, Jan. 2001.

[3] Z. H. Wei, B. J. Hu, E. J. Xia, and S. H. Lu, “A contention-free reporting scheme based MAC protocol for cooperative spectrum sensing in cognitive radio networks,” *IEEE Access*, vol. 6, pp. 38851–38859, Jul. 2018.

[4] Y. P. Fallah, F. Agharebparast, M. R. Minhas, H. M. Alnuweiri, and V. C. M. Leung, “Analytical modeling of contention-based bandwidth request mechanism in IEEE 802.16 wireless networks,” *IEEE Trans. Veh. Technol.*, vol. 57, no. 5, pp. 3094–3107, Sep. 2008.

[5] M. Rasheed, I. U. Din, M. Adnan, A. Tariq, S. Malik, and I. Syed, “ECM-MAC: An efficient collision mitigation strategy in contention based MAC protocol,” *IEEE Access*, vol. 9, pp. 62880–62889, Apr. 2021.

[6] T. Zhang and Q. Zhu, “EVC-TDMA: An enhanced TDMA based cooperative MAC protocol for vehicular networks,” *J. Commun. Netw.*, vol. 22, no. 4, pp. 316–325, Aug. 2020.

[7] V. Sevani, B. Raman, and P. Joshi, “Implementation-based evaluation of a full-fledged multihop TDMA-MAC for WiFi mesh networks,” *IEEE Trans. Mob. Comput.*, vol. 13, no. 2, pp. 392–406, Feb. 2014.

[8] H. Nam, “Interpolation-based SC-FDMA transmitter with localized resource allocation,” *IEEE Commun. Lett.*, vol. 14, no. 10, pp. 948–950, Oct. 2010.

[9] Y. Liu, G. Y. Li, Z. Tan, and H. Hu, “Noise power estimation in SC-FDMA systems,” *IEEE Wirel. Commun. Lett.*, vol. 4, no. 2, pp. 217–220, Apr. 2015.

[10] B. Zhi, W. Sainan, and H. Yigang, “A novel anti-collision algorithm in RFID for internet of things,” *IEEE Access*, vol. 6, pp. 45860–45874, Aug. 2018.

[11] Y. Cui and Y. Zhao, “Performance evaluation of a multi-branch tree algorithm in RFID,” *IEEE Trans. Commun.*, vol. 58, no. 5, pp. 1356–1364, May 2010.