Dynamic Adaptive Agent with Critical Movement Detection for the Next Generation of Spatial Reuse

Yoshima Syach Putri, Dion Tanjung, Dong-Hyun Kim, and Jong-Deok Kim*
Department of Information Convergence Engineering
Pusan National University
Busan, Republic of Korea
Email: {yoshima,tanjung.dn,dhkim1106,kimjd}@pusan.ac.kr

***Abstract*—Spatial Reuse (SR) is one of the spectrum efficiency mechanisms in 802.11ax, which has the goal to maximize the parallel transmission in wireless networks. Currently, SR adopts static Overlapping Basic Service Set with Packet Detect (OBSS/PD) threshold with BSS Color to adjust the sensitivity threshold. However, adjusting the proper threshold is challenging since an improper adjustment can cause a hidden/exposed node problem, thereby decreasing the throughput. This problem is worsened in a dynamic wireless environment where nodes move in unpredictable ways. This makes it more difficult for a static approach to determine its optimal configuration. To address this problem, we proposed a dynamic OBSS/PD threshold adjustment algorithm based on Multi-Armed Bandits. Furthermore, we introduce critical movement detection to identify when a node moves into or out of an interference zone. This detection is required to assist the MAB algorithm in generating a new distribution suitable for the new environment. Our simulation results show that a dynamic adaptive agent with critical movement detection can increase simultaneous transmission and achieve higher throughput in a dynamic STA mobility scenario.**

***Index Terms*—critical detection, IEEE 802.11ax, interference, multi armed bandits, OBSS/PD, spatial reuse**

I. Introduction

The rapid development in Wireless Local Area Networks (WLANs) has given a lot of impacts and significant improvement, especially for the dense network environments. One of the features introduced in 802.11ax is to enhance spatial and resource sharing. There are Orthogonal Frequency-Division Multiple-Access (OFDMA), Multi-User Multiple-Input-Multiple-Output (MU-MIMO), Spatial Reuse (SR), Target Wake Time (TWT), and Channel Bonding. In this paper, we only focused to improve the Spatial Reuse mechanism that has a goal to maximize parallel transmission in wireless networks. Currently, in the 802.11ax amendment, Spatial Reuse adopts two mechanisms, Overlapping Basic Service Set using Packet Detect (OBSS/PD) threshold with BSS Color and Spatial Reuse Parameter (SRP) [1]. The OBSS/PD threshold is used to ignore inter-BSS transmission, therefore it can increase the simultaneous transmission among OBSSs.

However, we cannot achieve the advantages of the OBSS/PD threshold if the threshold is not properly adjusted. The first concern is that of hidden and exposed node problems, which will typically occur if we only use static OBSS/PD threshold adjustment. To prevent this problem, we need an agent algorithm that dynamically adjusts the OBSS/PD threshold based on the environment. The second problem is the mobility of station (STA). This will happen in real-world scenarios, such as when we watch YouTube, make video calls, or stream while walking, which indicates that we have mobility as a STA that the Access Point (AP) cannot avoid. As a result, the AP has to be cognitive in order to adapt to its environments. If the AP can obtain information regarding the STA's mobility, the AP can provide appropriate treatment in dynamic scenarios in which the STA is continuously moving.

Currently, Reinforcement Learning (RL) is a subset of Machine Learning (ML) that is widely used to solve problems related to wireless networks environment, as RL's function is to adapt to the environment through defined actions. The first contribution in this paper is that we rely on Multi-Armed Bandits (MAB) that explicitly define the action based on arms in the agent and learn from the scenario. The output from MAB is a distribution that indicates the value that is closest to the true optimal reward in a given environment. As a consequence, if the environment changes, the current distribution may no longer be compatible with the new environment.

Therefore, our second contribution is to introduce a critical movement detection mechanism that can trigger the agent to reset the previous distribution value and re-learn based on the changed scenario. However, when we accommodate the movement detection, we have to disable the agent's ability to learn, as threshold learning has the potential to affect the width of the interference area. Hence, our third contribution is a mode switching algorithm based on MAB convergence that determines when the agent has sufficient information about the current situation. As a result, our agent is completely adaptable to its changing environment.

II. Related Works

Several approaches have been performed to adjust OBSS/PD threshold. Y. Kim [2] use random value to increase or decrease the threshold based on previous information from the successful flag. However, there is no margin or standard for the minimal or maximal random value. Selinis [3] implement Control OBSS/PD Threshold (COST) for adjusting OBSS/PD threshold based on the interference level and Received Signal Strength Indicator (RSSI) from the associated recipient. Ropitault [4] with a different approach claimed that his work, RSSI

*: Corresponding Author.

to OBSS/PD Threshold (RTOT), is the first algorithm to adjust OBSS/PD threshold. They use the beacon RSSI to dynamically compute the Carrier Sense Threshold (CST). It achieves 80% throughput higher than legacy but also 30% lower than legacy in several cases.

As mentioned in [2], [3], and [4], the authors introduce optimization based on heuristic algorithm. However, the optimization approach requires a lot of information which is hard to implement in the real environment such as location and the interference level of other APs. The optimization approach also needs to know the model well that means we should get the perfect knowledge about the model. If the model is inappropriate, then the optimization approach will be inaccurate as well. Unlike the optimization or heuristic approach, RL is a data-driven approach. It can learn from its environment and give a better solution for a problem where the model is unknown. Hence, RL will automatically adjust its solution, while the optimization approach may not adapt to environmental change because of its linear solution. Elif [10] also did the AI-Driven approach for CST adjustment using the number of collisions as one of their metrics evaluation. However, according to our best knowledge, it will be challenging to implement the solution in the real environment.

MAB in SR already perform in [5], [6], [7] to adjust Transmission Power Control (TPC) and CST. They use a static scenario to do the simulation. However, in the real environment, the location of the STAs might change continuously due to mobility. The STA mobility itself can be a problem if the STA moves towards high interference area or critical zone. Since it will decrease simultaneous transmission, we propose an algorithm to sequentially check whether the STA is moving into a critical zone or moving out from a critical zone. In addition, in paper [4] and [6], the author states that they haven't tried their approach in a dynamic environment because they know that the MAB algorithm will require additional information to adapt and provide a new distribution when the environment changes.

III. Proposed Method

A. OBSS/PD Adjustment based MAB

We select MAB as our OBSS/PD adjustment algorithm since the WLAN environment is stateless. MAB is a Reinforcement Learning algorithm that learns based on predefined actions called arms and receives rewards directly from the environment without knowledge of input or output state. There are three types of algorithm of MAB that most used by researchers [7], [8], [9]. Those types of algorithms are described as below:

1) Epsilon-Greedy (ε-Greedy): choose an action based on the probability of epsilon. If the random probability is greater than epsilon, then it will exploit the action. If the random probability is lower than epsilon, it will explore the action. Since it is simple to implement, ε-Greedy become mostly used MAB algorithm to solve many problems.

2) Upper Confidence Bound (UCB): choose an action based on the confidence level of each arm. This algorithm was developed to address the drawbacks of ε-Greedy, which is entirely based on random probability and epsilon values. The UCB maintains a confidence level for each arm in order to determine the frequency with which the arms are visited. The more frequently the arm is visited, the more assured it is. The arm that is less frequently visited will attempt to return at a later time. It is to ensure that the most exploited arm remains the proper arm over time. Or, during exploration, we discover that there is a more suitable arm than the one that has been most exploited over time.

3) Thompson Sampling (TS): choose an action based on the posterior sampling. This algorithm uses sampling from the posterior value and standard deviation. The results of the values taken from the sampling will be compared with each other. The highest sampling value determines the arms that will be selected in the next time step. This algorithm is also considered quite successful in terms of speed with UCB regarding the results to get the best arm to be exploited. Despite this, numerous researchers who used these two algorithms found that, depending on their issue scenarios, TS or UCB showed good results.

Later we will compare the results of each MAB algorithm, then we will use one of the algorithms that are considered the best to be combined with the next method.

B. Critical Movement Detection

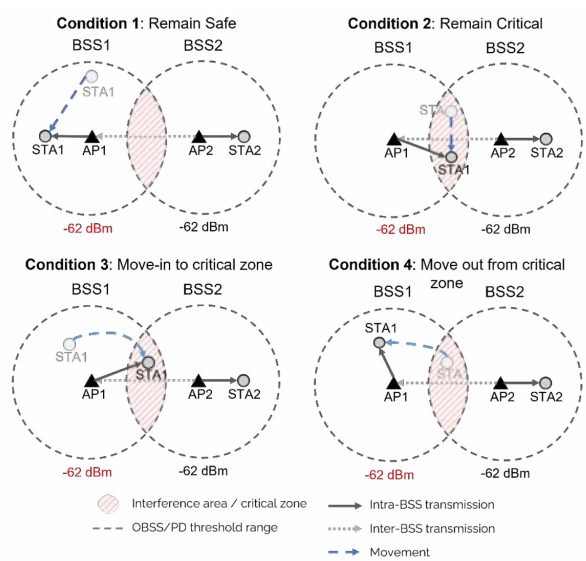

Figure 1. Type of movement

In a WLAN environment, there are two types of areas: critical zones and safe zones. A critical zone is an area of overlapping BSS that can cause a collision to the node that is located inside of that location. The safe zone is an area of WLAN scenario that which the node is out of the overlapping BSS or interference area depending on the transmission range from each BSS. The idea is come from the drawbacks of paper

[2] - [7] that does not consider the mobility of STAs. Since WLAN environments typically contain STAs that move from point to point, depending only on the MAB will result in poor performance as the STA's location changes. Because MAB will generate a distribution for scenarios that do not change.

As a result, changes to the scenario should also affect the MAB distribution. The proper action chosen prior to movement may not be suitable for the scenario presented after the STA has moved. However, not all movements have been considered critical movements. There are four conditions based on the STA's movement. In Fig. 1, condition 1 and 2 means that STA moving in the same area (remain safe zone or remain critical zone), while condition 3 and 4 means that STA moving in or out of the different areas (safe to critical or vice-versa).

Mobility becomes an issue when the STA moves to an area where data transmission is challenging or when it goes into a critical zone. Because when we are looking for the optimal OBSS/PD threshold in condition 1 and 2 in Fig. 1, the agent can easily learn to obtain the true optimal reward. However, in condition 3 and 4, each agent's learning will be dependent on the learning of the other agent, which means that learning will be tricky. Additionally, for condition 3 and 4, resetting the agent could be the optimal strategy because we will ignore the previous scenario and re-learn the new scenario, which will converge faster than learning with the previous value. As a result, we developed a critical movement detection algorithm that detects when a STA enters or exits a critical zone.

C. Mode Switching based on MAB Convergence

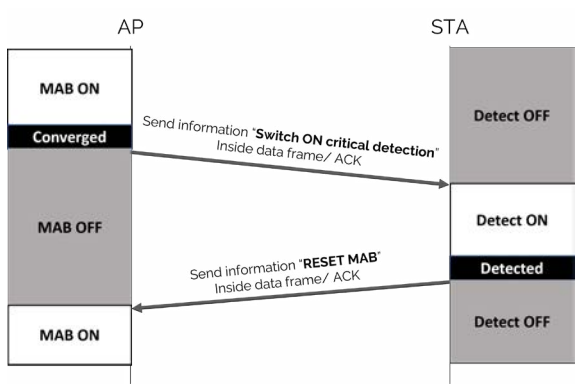

Figure 2. Mode Switching based on MAB Convergence

Mode switching based on MAB Convergence is a technique for stopping the Agent from learning when the result was considered as a convergence enough. This is performed not only for efficiency but also to avoid the detector from picking up the improper movement detection. As far as we know, MAB will always learn how to get the best OBSS/PD threshold, however, if this is performed continuously, the OBSS interference area or critical zone will change as well. As an outcome, critical movement detection will only be performed if the agent's state has sufficiently converged. This mechanism has two advantages: first, the agent does not have to learn again if it is considered convergence, and it minimizes detection errors.

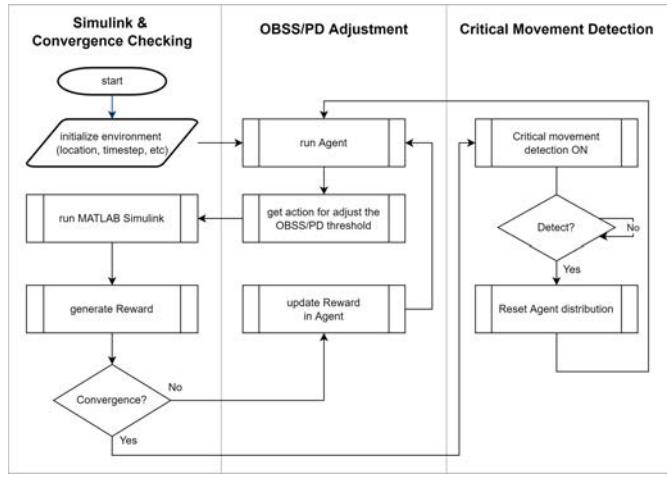

Figure 3. Algorithm Flowchart

The flow of the mode switching algorithm is shown in Fig. 2. The MAB algorithm is run on AP while the critical detection algorithm is run on STA. The change from the MAB algorithm to the critical detection algorithm occurs after the convergence of the MAB algorithm. While the change from the critical detection algorithm to the MAB algorithm occurs after a critical location change in the STA.

Table I
TABLE OF OPTIMAL REWARD IN 100 TIME STEPS (CONVERGENCE THRESHOLD 0.4)

Arm (dBm)	Scenario-1				Scenario-2			
	Percentage of Arm chosen (%)		AVG throughput per arms (Mbps)		Percentage of Arm chosen (%)		AVG throughput per arms (Mbps)	
	BSS1	BSS2	BSS1	BSS2	BSS1	BSS2	BSS1	BSS2
-62	0.06	0.06	2.52	1.84	0.03	0.06	3.12	1.84
-67	0.02	0.06	0.84	2.08	0.03	0.04	0	0
-72	0.36	**0.78**	3.67	**3.97**	0.29	0.08	3.28	2.16
-77	**0.44**	0.06	**3.92**	3.8	0.07	0.12	3.26	2.88
-82	0.12	0.04	3.72	3.00	**0.59**	**0.70**	**3.29**	**3.11**

Here, a threshold value is required to determine whether the agent is considered to be convergent or not. We make an observation about the total probability that the arm with the highest reward is chosen in a short time step. We discover that when the threshold is set to 0.4, the best arm can be selected immediately, as illustrated in Table I for two scenarios illustrated in Fig. 4. However, we will use a value close to ideal in the experiment, specifically 0.8, because the maximum probability we get for time step 50th is 0.78. We want to rapidly determine the optimal threshold so that the agent does not require a long period of time to be considered convergent.

D. Algorithm Design

Fig. 3 depicts the set of algorithms we proposed. We begin with Simulink and Convergence Checking by initializing the WLAN environment scenario, followed by running the agent in MATLAB. The WLAN simulation will generate a reward

and determine whether the agent's reward value and arm information have converged using OBSS/PD adjustment. If not, the agent will automatically update the reward and move on to the next time step to run the agent and receive action from the OBSS/PD Adjustment process. When convergence is reached, the agent turns off the OBSS/PD threshold adjustment algorithm and activates critical movement detection. This means that the agent no longer needs to adjust the threshold to ensure that the critical zone remains constant at the next time step.

In the Critical Movement Detection process, the STA will receive and process the signal from the inter-BSS transmission. Using a decision tree algorithm, STA will determine whether it is significantly moved or not. It starts when the STA detects an inter-BSS signal. Then it determines whether the previous inter-BSS signal was greater than the OBSS/PD threshold or not. If yes, it means that the STA was in the critical zone in the previous state. Otherwise, the STA is in a safe zone. After that, it will determine whether the current inter-BSS signal received by STA is greater than the OBSS/PD threshold. If yes, it will compare the current state to the previous state.

There are four primary evaluations based on comparing previous and current inter-BSS receiving signal. These include entry into a critical zone, exit from a critical zone, remaining safe, and remaining critical. As mentioned in the previous chapter, the critical condition in this section is the detection of critical zone entry and exit. And whenever the STA detects either of those two conditions, the STA will issue a "reset" command to the agent. The agent in AP will reset the distribution and restart the MAB algorithm if it receives a "reset" command from the STA on the same BSS.

The process will be repeated until the STA achieves convergence and detects significant movement. Convergence time will be short, as the agent will always have data to send to the STA in the case of streaming, video calls, and real-time high data transmission. A 0.1-second timestep that contains sufficient information from the bulk data transmission accelerates the algorithm's convergence. This will be demonstrated through experimental results in the following section.

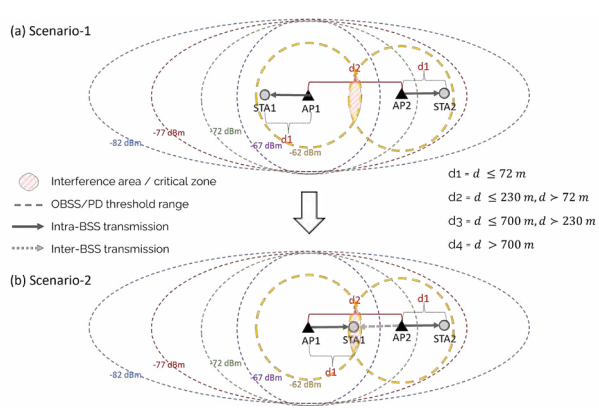

Figure 4. Mobility scenario (a) Scenario-1; (b) Scenario-2

IV. Experiments and Results

To run WLAN scenario simulations, we use MATLAB R2021b and Python 3.9, which are compatible with the MATLAB Simulink version. The scenario that we created is scenario-1, as illustrated in Fig. 4(a), and the STA1 will enter the critical zone, as illustrated in Fig. 4(b). The AP may have a different OBSS/PD threshold for each STA, as the SR mechanism operates only when AP has data to transmit to the STA. Therefore, in this paper, we consider only the scenario in which each BSS has a single STA. The distance between each AP-STA is 72 m, which is sufficient to cover the STA of its BSS. The OBSS/PD threshold is highly correlated with Transmission Power (TxP) and Receiver Signal Sensitivity Range (RSSI). According to the paper [4], we can calculate the TxP based on OBSS/PD threshold that we choose. This algorithm is to minimize the interference range for another BSS. The algorithm (1) demonstrates a correlation between TxP and OBSS/PD threshold. Additionally, in these experiments, we employ the Free Space Path Loss model.

$$TxP = OBSS\_PD_{min} + TxP_{ref} - OBSS\_PD \quad (1)$$

Table II
Table of Optimal Reward

OBSS/PD threshold (dBm)		Scenario-1			Scenario-2		
		Throughput (Mbps)		Mean (Mbps)	Throughput (Mbps)		Mean (Mbps)
BSS1	BSS2	BSS1	BSS2		BSS1	BSS2	
-62	-62	5.64	5.52	5.58	0	5.52	2.76
-62	-67	2.40	6.24	4.32	0	6.24	3.12
-62	-72	0	6.24	3.12	0	6.24	3.12
-62	-77	0	6.24	3.12	0	6.24	3.12
-62	-82	0	6.24	3.12	0	6.24	3.12
-67	-62	6.24	0	3.12	3.6	0	1.80
-67	-67	6.20	5.40	5.80	0	5.76	2.88
-67	-72	1.32	6.24	3.78	0	6.24	3.12
-67	-77	0	6.24	3.12	0	6.24	3.12
-67	-82	0	6.24	3.12	0	6.24	3.12
-72	-62	6.24	0	3.12	3.72	0	1.86
-72	-67	6.24	0	3.12	3.96	0.12	2.04
-72	**-72**	**5.88**	**5.76**	**5.82**	0	5.28	2.64
-72	-77	3.60	5.40	4.50	0	5.40	2.70
-72	-82	3.12	2.88	3.00	3.12	3.12	3.12
-77	-62	6.24	0	3.12	3.72	0	1.86
-77	-67	6.24	0	3.12	3.72	0	1.86
-77	**-72**	5.64	3.36	4.50	**3.24**	**3.36**	**3.30**
-77	-77	3.36	2.88	3.12	3.12	3.12	3.12
-77	-82	3.36	2.88	3.12	3.12	3.12	3.12
-82	-62	6.24	0	3.12	3.72	0	1.86
-82	-67	6.24	0	3.12	3.72	0	1.86
-82	-72	3.84	2.16	3.00	3.84	2.16	3.00
-82	-77	3.36	2.88	3.12	3.36	2.88	3.12
-82	-82	3.36	2.88	3.12	3.12	3.12	3.12

A. The Agent with Optimal Reward

To find out whether the results of our experiment have reached the optimal reward, we have done collecting distribution data for each selected OBSS/PD threshold. The results of the combined OBSS/PD threshold for the two BSS are attached in Table II. For example, if BSS1 selects -62 dBm and BSS2 selects -82 dBm for Scenario-1 in Fig. 4, BSS1's throughput will be 0 Mbps and BSS's will be 6.24 Mbps. As shown, BSS2's transmission range covers the whole of the BSS1 node. In that case, BSS1 will experience an exposed

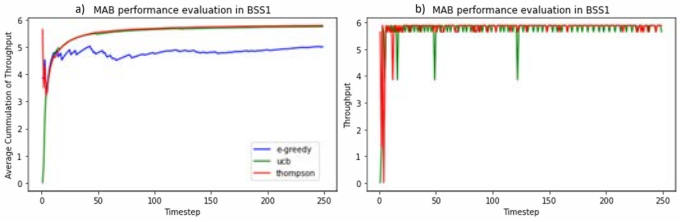

Figure 5. Performance Evaluation Single-Agent using MAB algorithm

node problem in AP1 and a hidden node problem in STA1, as STA1 will simultaneously receive both transmissions.

As a result, it is shown that the optimal OBSS/PD threshold for obtaining the true optimal reward should be both BSS with -72 dBm in Scenario-1 and BSS1 and BSS2 with -77 dBm and -72 dBm in Scenario-2, respectively. In this section, before doing dynamic multi-agent, we tested how dynamic single-agent works (on BSS1 only) to ensure that the agent with the specified MAB can select the OBSS/PD threshold properly. Later, we will compare the results to determine which algorithm converges to the optimal threshold the fastest. With Scenario-1, we set a static OBSS/PD threshold of -72 dBm on BSS2 and observe how BSS1 chooses the proper OBSS/PD threshold for it.

The results in Fig. 5 a) indicate that all algorithms are capable of achieving convergence, with BSS1 selecting OBSS/PD -72 dBm as the best case for this scenario. However, because ε-Greedy's performance is inferior to that of UCB and Thompson, we will compare the throughput in each time step only to those two algorithms. Then in Fig. 5 b), we discovered that Thompson Sampling converges faster than UCB. As a result, we choose Thompson Sampling as the algorithm for adjusting the OBSS/PD thresholds in Agent. This also demonstrates that a single agent is capable of finding the optimal reward when converged in less than 100-time steps.

Figure 6. Performance Evaluation Multi-Agent without and with Distribution Reset

To demonstrate that the Agent's Reset has an effect on threshold selection, we create a random scenario in which STA moves from a random point in the safe zone to a random point in the critical zone at time step 500. Furthermore, we will manually reset the agent to identify the impact of a correct reset in threshold learning. This is to illustrate that Agents with MAB do require a new distribution upon a significant change in location. And the results are shown in Fig. 6, where the agent with distribution reset during critical mobility achieves higher throughput and convergence than the agent without distribution reset.

B. The Agent with Auto-Reset Detection

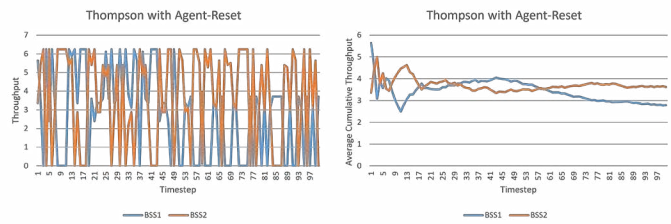

Figure 7. Combining agent and auto-reset detection results in a new convergence problem.

This section outlines the result of the Scenario in Fig. 4 when multi-agent learning and critical movement detection are used to reset the learning process. This is quite intriguing to observe because, as explained previously, the critical zone will always vary depending on the transmission range of each BSS. As a result, when the two proposed methods are combined, they will be highly sensitive. And the third experiment illustrated in Fig. 7 illustrates that this is truly the case, as the throughput at each time step becomes highly unstable and can never converge. This is because the MAB agent learns every time it detects movement into or out of the critical zone by adjusting the OBSS/PD threshold. As a result, the critical zone will vary with the OBSS/PD threshold selected at each time step.

As a result, we proposed a mode switching algorithm to address the issue that has occurred in the process of combining agent and critical movement detection. The goal of this algorithm is to switch the agent and detection processes so that they do not run simultaneously. This fourth experiment is designed to ensure that the OBSS/PD threshold adjustment and critical movement detection run as efficiently as possible. Where OBSS/PD threshold adjustment occurs when the agent does not have sufficient information about the environment and terminates when the agent converges on the best appropriate threshold. Additionally, critical movement detection will occur only after the agent has converged and notified STA of their purpose to detect their own mobility actively.

We run the scenario with 500-time steps and the results are given in Fig. 8 which compares the effect of a 40% and an 80% convergence threshold on the agent's convergence value. The result shows that both BSS chose an OBSS/PD threshold of -72 dBm as their convergence value in Scenario-1, with a throughput of 5.88 Mbps for BSS1 and 5.76 Mbps for BSS2. It is coherent with Scenario-1's optimal reward, as specified in Table II, in that the proving agent can achieve a mean throughput of 5.82 Mbps.

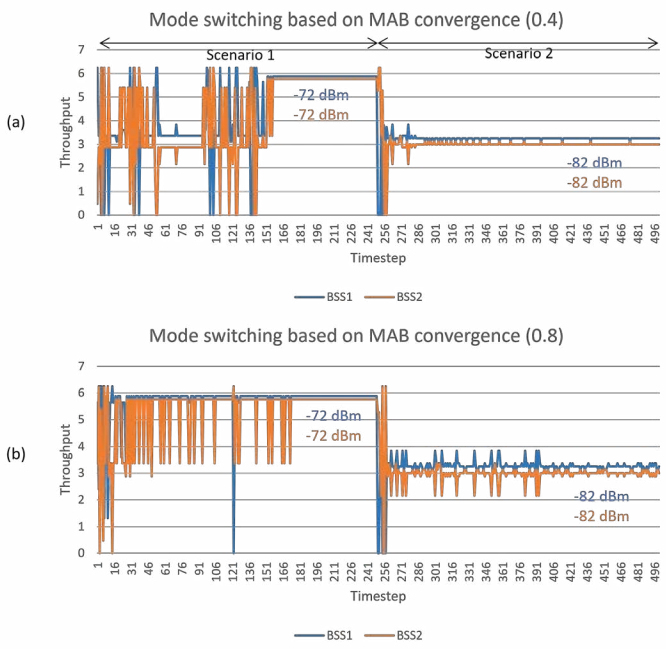

Figure 8. Performance Evaluation (Throughput) of mode switching algorithm based on MAB convergence with (a) convergence threshold = 40%, (b) convergence threshold = 80%

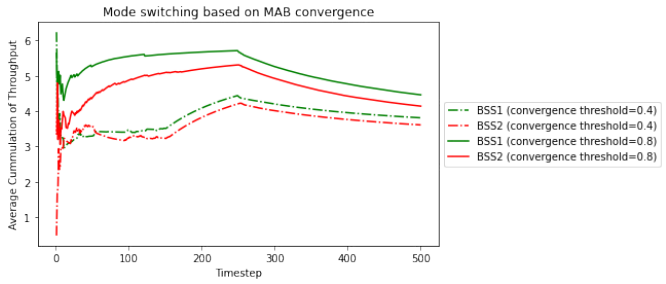

Figure 9. Performance Evaluation (Average Cumulative Throughput) of mode switching algorithm based on MAB convergence with convergence threshold 40% and 80%

However, in Scenario-2, where the agent should choose -77 and -72 dBm for BSS1 and BSS2 based on Table II, the agent fails to provide it, as illustrated in Fig. 8. Each agent choose -82 dBm OBSS/PD threshold with a throughput of 3.12 Mbps. Since this scenario contains a critical zone in which STA1 is located between AP1 and AP2, within the transmission range of 1 dBm, it can not do parallel transmission. Here, we want to emphasize that the MAB algorithm itself still has flaws as a result of each agent's selfish selection. However, when the MAB results are combined with our proposed algorithm, the result is still superior to the static OBSS/PD threshold in a static scenario. As we can see, the agent is successfully maximizing throughput, as the difference between the mean throughput of Scenario-2 and the optimal reward of 3.30 Mbps is only 0.17 Mbps.

However, as illustrated in Fig. 9, the throughput of the two agents with the convergence threshold is 80% greater than the throughput of the two agents with a convergence threshold is 40%. This is because, as illustrated in Fig. 8, the agent made an error in the 50th time step by selecting the incorrect arm after determining that it had already reached the 40% threshold. Right after the agent chooses the arm, the more confident the agent's arms distribution is, the more it recognizes that choosing that arm was a mistake. Thus, the agent will explore again in the next time step until it reaches the 40% threshold. As a result, the optimal value for indicating that the agent has already converged is 80% of the convergence threshold.

V. Conclusion

In this paper, we evaluate that using a dynamic adaptive agent with critical movement detection will solve the problem where we can maximize the OBSS/PD threshold adjustment for dynamic scenarios or environments. Furthermore, the results show that this algorithm is capable of generating optimal rewards that maximize the benefits of spatial reuse in a WLAN environment. Because even in challenging situations where parallel transmission is not possible, this algorithm can maximize throughput.

In order to test this algorithm in various combinations of scenarios, we leave it to future work. For example, changing the density of each BSS and the location of the multiple critical STAs. Furthermore, we can determine the possibility of optimizing the mode switching algorithm. In addition, another fascinating approach could be used in combination with this algorithm, such as optimizing resource allocation using information gathered from the STA's movement detection.

VI. Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2020R1I1A3065947).

References

[1] F. Wihelmi, S. B. Munoz, C. Cano, I. Selinis, B. Bellalta, "Spatial Reuse in IEEE 802.11ax WLANs," Computer Communications, Vol.170, 2021.
[2] Y. Kim, G. Kim, T. Kim, W. Choi, "Transmission Opportunity-Based Distributed OBSS/PD Determination Method in IEEE 802.11ax Networks," ICAIIC, pp. 469, 2020.
[3] I. Selinis, K. Katsaros, S. Vahid and R. Tafazolli, "Control OBSS/PD Sensitivity Threshold for IEEE 802.11ax BSS Color," 2018 IEEE 29th Annual International Symposium on PIMRC, pp. 1-7, 2018.
[4] T. Ropitault, "Evaluation of RTOT algorithm: A first implementation of OBSS PD-based SR method for IEEE 802.11ax," CCNC, pp. 1-7, 2018.
[5] F. Wihelmi, C. Cano, G. Neu, B. Bellalta, A. Jonsson, S. B. Munoz, "Collaborative Spatial Reuse in wireless networks via selfish Multi-Armed Bandits," Ad Hoc Networks, 2019.
[6] F. Wihelmi, S. B. Munoz, B. Bellalta, C. Cano, A. Jonsson, G. Neu, "Potential and pitfalls of Multi-Armed Bandits for decentralized Spatial Reuse in WLANs," Journal of Network and Computer App, 2019.
[7] M. Carrascosa, B. Bellalta, "Multi-Armed Bandits for Decentralized AP selection in Enterprise WLANs," Computer Communications, Vol.159, 1 June 2020.
[8] M. Yun, D. Pang, A. Latif, "The Multi-Armed Bandit Problem: An exploration of epsilon greedy and UCB1", accessed 10 November 2021, https://cse442-17f.github.io/LinUCB
[9] Z. Elaine, "Comparing Multi-Armed Bandit Algorithms on Marketing Use Cases," accessed 28 December 2021, towardsdatascience.com.
[10] A. Elif, C. Berk, "FSC: Two Scale AI-Driven Fair Sensitivity Control for 802.11ax Networks," IEEE Global Communications Conference, 2020.