symmetry

Article

An Enhanced Active Access-Point Configuration Algorithm Using the Throughput Request Satisfaction Method for an Energy-Efficient Wireless Local-Area Network

Bin Wu [1], Nobuo Funabiki [1,*], Dezheng Kong [1], Xuan Wang [1], Taishiro Seto [1] and Yu-Cheng Fan [2]

[1] Graduate School of Environmental, Life, Natural Science and Technology, Okayama University, Okayama 700-8530, Japan; ppxf92cc@s.okayama-u.ac.jp (B.W.); pxep6n2q@s.okayama-u.ac.jp (D.K.); ptqd3isy@s.okayama-u.ac.jp (X.W.); prhh25mt@s.okayama-u.ac.jp (T.S.)
[2] Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan; skystar@ntut.edu.tw
* Correspondence: funabiki@okayama-u.ac.jp

Abstract: *Wireless Local-Area Networks (WLANs)*, as a popular internet access solution, are widely used in numerous places, including enterprises, campuses, and public venues. As the number of devices increases, large-scale deployments will cause the problem of dense wireless networks, including a lot of energy consumption. Thus, the optimization of energy-efficient wireless AP devices has become a focal point of attention. To reduce energy consumption, we have proposed the *active access-point (AP) configuration algorithm* for WLANs using APs with a dual interface. This uses the greedy algorithm combined with the *local search* optimization method to find the minimum number of activated APs while satisfying the minimum throughput constraint. However, the previous algorithm basically satisfies only the average throughput among the multiple hosts associated with one AP, wherein some hosts may not reach the required one. In this paper, to overcome this limitation, we propose an *enhanced active AP configuration algorithm* by incorporating the *throughput request satisfaction method* that controls the actual throughput at the target value (*target throughput*) for every host by applying *traffic shaping*. The *target throughput* is calculated from the single and concurrent communicating throughput of each host based on channel occupancy time. The minimum throughput constraint will be iteratively adjusted to obtain the required *target throughput* and achieve the fair throughput allocation. For evaluations, we conducted simulations using the *WIMNET simulator* and experiments using the testbed system with a *Raspberry Pi 4B* for APs in four topology cases with five APs and ten hosts. The results show that the proposed method always achieved the required minimum throughput in simulations as well as in experiments, while minimizing the number of active APs. Thus, the validity and effectiveness of our proposal were confirmed.

Keywords: energy-efficient WLAN; IoT; active AP configuration algorithm; throughput request satisfaction method; throughput control; traffic shaping

Citation: Wu, B.; Funabiki, N.; Kong, D.; Wang, X.; Seto, T.; Fan, Y.-C. An Enhanced Active Access-Point Configuration Algorithm Using the Throughput Request Satisfaction Method for an Energy-Efficient Wireless Local-Area Network. *Symmetry* **2024**, *16*, 1005. https://doi.org/10.3390/sym16081005

Academic Editors: Mohammed H. Alsharif and Md. Farhad Hossain

Received: 1 July 2024
Revised: 31 July 2024
Accepted: 2 August 2024
Published: 7 August 2024

1. Introduction

As the most popular Internet access method, *IEEE 802.11* protocols in *Wireless Local-Area Networks (WLANs)* have been applied in various scenarios, such as the *Internet of Things (IoT)* and *Wireless Sensor Networks*, as well as densely populated user areas like offices or schools, due to their versatility, convenience, and cost-effectiveness [1–3]. In WLANs, a host connects wirelessly with an access point (AP), providing greater extensibility and flexibility compared to wired LANs [4].

As the number of users and devices grows, the congestion on WLANs increases under the limited number of available channels. Consequently, the issue of dense WLAN environments has been raised to address the challenges of APs in supporting a large number of users. To avoid performance degradation, the network configuration of a

WLAN, including the number and coverage of active APs, their channel assignments, locations, and transmission power, should be properly optimized according to traffic demands [5,6].

Considering the need to reduce the cost of aging devices and power consumption, particularly in IoT application systems demanding long lifespans and low-cost deployments, an increasing number of researchers have conducted studies on *efficient energy-saving wireless network design methods* [7]. These methods include energy-efficient resource allocation [8,9] and *Media Access Control (MAC)* protocol improvements [10]. Their goal is to extend the operational lifespans of IoT devices, reduce overall power usage, and ensure cost-effective and sustainable networks over time.

To address energy-saving issues and performance optimizations in dense network environments, we previously proposed the *active AP configuration algorithm* using dual interfaces [11]. This algorithm enables dynamic optimization of the network configuration by activating or deactivating AP devices while considering the AP setup optimization regarding the channel assignment and the host association. This study computes the total transmission capacity potentially available in the network. Then, it determines whether the minimum throughput constraint is satisfied on average and optimizes the number of activating APs using a *local search method*.

However, when multiple hosts are associated with the same AP and are located at different distances from it, the issue of the throughput *unfairness* or *insufficiency* will arise due to the interference among them. A host enjoys higher received signal strength (RSS) when it is located closer to its AP compared to one situated farther away [12]. Therefore, the *throughput unfairness/insufficiency* problem may appear among the hosts and, due to this advantage of RSS, the result will be a larger *TCP congestion window size* and a higher *modulation and coding scheme (MCS)* at the transmitting packets, which will create higher throughput.

To solve this problem, we have studied the *throughput request satisfaction method* [13,14]. It calculates the channel occupancy time and the *target throughput* for each associated host from the measured *single throughput* and *concurrent throughput* for it. The *single throughput* is measured when only one host is communicating with the AP. The *concurrent throughput* is measured when all the hosts are communicating with the APs. By controlling the outgoing data rates at the AP using *traffic shaping* at the *target throughput*, the fair or requested throughput is achieved for each host.

In the previous active AP configuration algorithm, only the average throughput among the hosts associated with the same AP is satisfied. If a host is far from the AP, it may not achieve the required minimum throughput. To overcome this deficiency, in this paper, we enhance the *enhanced active AP algorithm* by incorporating the throughput request satisfaction method. This enhancement consists of the following steps:

1. Applying the previous active AP configuration algorithm to find the network configuration. The *single throughput* and *concurrent throughput* for every host is estimated by the *throughput estimation model* [15].
2. Calculate the *target throughput* for the fair throughput to every host using the throughput request satisfaction method.
3. If this *target throughput* does not satisfy the minimum host throughput, the tentative minimum host throughput is increased by a constant, and the *active AP configuration* is applied in step 1.

Our design of an energy-efficient WLAN fair throughput allocation algorithm can be achieved by exploring network symmetry and the fairness of resource allocation. This can improve network efficiency by optimizing resource utilization, reducing energy consumption, and ensuring fair access among different devices. The contributions of this proposal for energy saving are summarized as follows:

- The number of active APs consuming energy is minimized by the active AP configuration algorithm.

- Under the adoption of dual interface devices for APs, our algorithm can both find the minimum number of active APs and allow any host to enjoy the minimum throughput.
- The enhanced algorithm can achieve fair throughput allocation and satisfy the minimum throughput constraint among the hosts. Meanwhile, the number of active APs will not increase in most cases.

For evaluations, we verified the validity of the proposal through simulations using the *WIMNET simulator* [16] and through experiments using the testbed system where a *Raspberry Pi 4B* with dual interfaces was used for the AP. For every channel of the AP, *channel bonding* is used, since it basically provides higher throughput [17,18]. In both evaluations, four network topology cases were considered. The results from the simulations show that the *minimum throughput constraint* was satisfied in every case by the proposal where all the hosts associated with the same AP enjoy the same throughput. Then, the results from the experiments show that it was satisfied in all cases. Thus, the validity and effectiveness of the proposal are confirmed.

The remainder of this paper is structured as follows. Section 2 introduces the literature review. Section 3 introduces the preliminary work of this study. Section 4 proposes the enhanced active AP configuration algorithm using the throughput request satisfaction method for an energy-efficient WLAN. Section 5 evaluates the proposal through simulations and experiments. Finally, Section 6 concludes this paper with future work.

2. Related Work in the Literature

In this section, we introduce related work in the literature. The studies are classified into *throughput control/allocation* issues and *energy-saving WLAN design* issues.

2.1. Throughput Control/Allocation

In [19], Mao addressed the problem of joint AP association and transmission time allocation in densely deployed multi-rate WLANs using the *time-sharing* MAC protocol. The issue was framed as an NP-hard *single non-zero programming (SNZP)* optimization problem with a goal of achieving *proportionally fair (PF)* throughput. To address this, the author introduced two innovative algorithms: *SNZP relaxation (SNZPR)* and *iterative SNZPR (iSNZPR)*. Moreover, for dynamic network conditions, a distributed joint admission, AP association, and transmission time allocation (DAAA) algorithm was developed. The performances of these algorithms were compared with existing algorithms through numerical analysis. In contrast, we use *traffic shaping* to balance the concurrent communicating throughput based on *channel occupancy time* for each host to achieve fair throughput allocation. At the same time, our algorithm can improve the host association for minimum activated APs and achieve an energy-efficient network.

In [20], Kim et al. addressed the issue of per-station fairness in uplink WLANs by tackling imbalances in access and outage probabilities. They utilized an enhanced distributed coordination function (DCF) combined with a hybrid automatic repeat request (HARQ) protocol, specifically HARQ with Chase combining (HARQ-CC). They also proposed a new Markov chain model for performance analysis, providing closed-form expressions for system throughput, delay, and outage probability. Their results demonstrated that the algorithm ensured near-perfect per-station fairness and improved overall system performance.In contrast, our proposal does not require modifications to the *MAC scheme*, making it easier to implement in real-world scenarios. We achieve fair allocation by controlling incoming and outgoing traffic through traffic shaping, setting the data rate to represent the *channel occupancy time*.

In [21], Chen et al. introduced a *Target Wake Time (TWT)* scheduling scheme to manage throughput in the *IEEE 802.11ax* protocol. This scheme addresses OFDMA-based multiuser transmissions by minimizing resource conflicts among sleeping stations during each beacon interval. Their work emphasizes strategies to avoid collisions and allocate throughput efficiently while saving power through an innovative broadcast *TWT* approach. This approach enables the AP to specify the *Target Beacon Transmission Time (TBTT)* for each station that requests *TWT*, thereby improving overall throughput. The effectiveness of the *TWT* scheme was assessed using simulations, with conclusions drawn from the results.In contrast, our method is applicable to any WLAN protocol situation, achieving fair throughput allocation. For certain devices, especially in IoT scenarios where only a few currently support the *802.11ax* protocol, our approach is designed for practical applications and can be effectively implemented and rapidly deployed.

In [22], Khorov et al. presented a centralized approach called *SEBRA (SAND-Enabled Bitrate and Resource Allocation)* designed to improve network-assisted video streaming over wireless networks. *SEBRA* functions on access points (APs) to effectively control the distribution of video bitrates among clients and the allocation of channel time. The primary goal of this algorithm is to optimize the duration of channel occupancy in line with the bitrate requirements of each client. This problem was formulated as an NP-hard issue, for which the authors employed *heuristic algorithms* to find optimal solutions regarding channel time allocation and resource management.In contrast, our approach can adapt to different requirements by adjusting the traffic type. It is applicable not only to high-speed traffic such as network video but also to low-latency, low-traffic IoT environments.

In [23], Yagi et al. presented two novel control strategies that leverage frame aggregation in *IEEE802.11n/ac* to enhance throughput and fairness across multiple WLANs in high-density environments. These strategies involved the dynamic adjustment of *transmission frequency* and *frame aggregation size* to reduce error probability. Their effectiveness was validated through simulations conducted with the network simulator *NS-3* [24].In contrast, our approach does not necessitate changes to the MAC scheme, simplifying its implementation in practical applications. We ensure fair allocation by managing incoming and outgoing traffic via traffic shaping, where the data rate is used to indicate the channel occupancy time. At the same time, the experiments have verified the effectiveness of our proposal.

In [25], Obata et al. proposed a switching method for *Media Access Control (MAC)* to enhance throughput fairness between WLAN systems under adjacent channel interference. The method dynamically switched between *CSMA/CA* and *semi-active contention window adaptation (SACA)* based on measured throughput, utilizing the capture effect to improve performance. Simulations using *NS-3* demonstrated the method's effectiveness in maintaining fairness and throughput in densely deployed environments.In contrast, we ensure fair allocation by adjusting the channel occupancy time without needing to modify the MAC scheme, making it easier to deploy.

2.2. Energy-Saving WLAN Design

In [26], Kong et al. proposed a technique for optimizing AP deployment using the *multi-objective particle swarm optimization (MOPSO) algorithm*. They began by evaluating the performance of a single AP through random geometry theory to determine the necessary number of APs for the WLAN based on user service demands. Next, the MOPSO algorithm was used to identify the optimal positions and transmit power levels for the APs. Finally, a greedy algorithm was implemented to remove any redundant APs.In contrast, our proposed algorithm generates an initial solution based on the *greedy algorithm* and optimizes host association through *local search*. This ensures that the throughput requirements of all hosts are met while minimizing the number of active APs. Additionally, the channels configured for the APs are optimized to reduce interference.

In [27], Umar et. al proposed a method for using phone user clustering (PUC) to group users based on proximity and channel conditions, optimizing resource allocation and minimizing interference. This method, combined with *hybrid multiple access (H-MA)* techniques that integrate *non-orthogonal multiple access (NOMA)* with *orthogonal multiple access (OMA)*, aims to efficiently utilize radio spectrum and power. Unmanned aerial vehicles (UAVs) are incorporated to provide flexible and targeted coverage, further improving throughput and energy efficiency. This approach effectively balances resource allocation, reduces power consumption, and enhances overall network efficiency, making it a promising solution for future 6G networks. In contrast, our proposal focuses on indoor WLAN optimization, and we use dual-interface devices to reduce the activated APs to achieve an energy-efficient network.

In [28], Blobel et al. proposed a method for achieving energy efficiency in WLANs by integrating *wake-up receivers (WuRxs)*. This method modified the standard *IEEE 802.11* protocol to include a wake-up signal, allowing devices to stay in low-power mode until needed, thus saving energy. Their approach was fully compatible with existing WLAN standards, enabling gradual deployment. Experimental results using a hardware prototype and simulations showed significant energy savings, low delays, and improved performance compared to traditional duty-cycling techniques.In contrast, our algorithm optimizes AP activation and connection allocation in the current network, achieving the minimum number of APs while ensuring the minimum throughput for hosts. This method does not require complex protocol modifications and can be directly deployed in practical applications.

In [29], Ali et al. proposed the *greenMAC* protocol, which enhances WLAN energy efficiency using *Q-learning*, a reinforcement learning technique. This protocol optimizes the energy-saving process by adjusting the *power-saving mode (PSM)* of WLAN devices based on channel congestion observations. By utilizing *Q-learning* to evaluate channel collision probabilities, the *greenMAC* protocol dynamically decides whether a device should enter sleep mode in order to reduce unnecessary energy consumption while maintaining network throughput. This method resulted in significant energy savings for WLANs, especially in high-density environments, without sacrificing performance.In contrast, our algorithm focuses on optimizing AP activation and connection allocation within the existing network. It aims to minimize the number of APs needed while guaranteeing the minimum required throughput for hosts. This approach avoids complex protocol modifications, allowing for straightforward deployment in real-world scenarios.

In [30], Dong et al. proposed the *ES-MPTCP* algorithm to optimize energy consumption in WLANs using *Multipath TCP (MPTCP)*. The *ES-MPTCP* algorithm balances the energy consumption and network throughput by dynamically selecting the optimal sub-flows based on current network conditions. This approach reduced the energy usage by 16.2% and increased the throughput by 13.6%, achieving higher energy efficiency compared to existing methods.In contrast, our algorithm utilizes dual-interface APs, offering two different frequency bands and providing more options for host allocation. This approach effectively reduces the number of active APs and minimizes interference in the network, thereby enhancing overall network performance.

In [31], Karmakar et al. presented a method to improve WLAN energy efficiency through the *GreenAP* framework. This method minimized energy consumption by optimizing the AP association and channel width selection, activating only the necessary APs and using energy-efficient transmissions within active *BSSs*. The approach saved significant energy without compromising the network performance by dynamically managing the AP activation and channel bonding. In contrast, our algorithm also focuses on optimizing AP activation and channel allocation. Additionally, it ensures rational planning of throughput allocation for each host, enabling all hosts to enjoy fair throughput sizes while achieving energy efficiency.

In [32], Qiu et al. proposed an energy-efficient method for dense WiFi networks based on *IEEE 802.11ax*. This method achieved energy savings by jointly optimizing the AP placement and *power-channel-resource unit (RU)* assignments. The objectives were to minimize the number of APs, fulfill user throughput requirements, and ensure AP fault tolerances. The authors utilized *orthogonal frequency division multiple access (OFDMA)* to divide the wireless spectrum into time-frequency resource units and designed a heuristic algorithm to find high-quality solutions within polynomial time complexity. Simulation results showed that their algorithm effectively reduced the number of APs and enhanced network performance [32].In contrast, our method does not require a specific communication protocol, making it widely applicable. The primary goal of our algorithm is to achieve an energy-efficient network by reducing the number of activated APs. The algorithm provides optimal AP activation and host connection schemes, and it incorporates a throughput fairness allocation method, allowing for straightforward deployment in practical applications.

3. Preliminary Work

In this section, we discuss our preliminary work related to this study. In addition, we review our previous active AP configuration algorithm and the throughput request satisfaction method, and we demonstrate an analysis of potential throughput insufficiency and fairness issues that may arise with hosts in the previous algorithm.

3.1. Throughput Estimation Model

First, we introduce the throughput estimation model, which provides the foundation for throughput calculations in algorithm simulations. This model estimates the throughput between an AP and a host in a WLAN network.

3.1.1. Received Signal Strength Estimation

In our study, we use the *log-distance path-loss model* to estimate the received signal strength at the destination node [33]. The *Euclidean distance d* (in meters) for each link or the AP/host pair is determined using the following formula:

$$d = \sqrt{(AP_x - H_x)^2 + (AP_y - H_y)^2}, \tag{1}$$

where AP_x and AP_y represent the x and y coordinates of the access point, and H_x and H_y represent the x and y coordinates of the user host, respectively.

Then, the estimated received signal strength, denoted as Rss (in $-dBm$), at the host is as follows:

$$Rss = Rss_{1m} - 10\alpha \log_{10} d - \sum_k n_k W_k. \tag{2}$$

In this paper, we define Rss_{1m} as the received signal strength from the access point (AP) to the host when they are one meter apart with no obstacles in between. The path loss exponent is represented by α. The variable n_k indicates the count of type k obstacles or walls along the path between the AP and the host, while W_k denotes the signal attenuation in dBm for each obstacle type k (with a range from 1 to 6). It is important to note that a building can have multiple types of walls. In this study, we consider six different types of obstacles: W_1 for corridor walls, W_2 for partition walls, W_3 for intervening walls, W_4 for glass walls, W_5 for elevator walls, and W_6 for doors, as noted in previous studies [11].

3.1.2. Throughput Estimation

Based on the RSS calculation, we established a functional relationship between RSS and throughput using experimental data. This estimation can provide us with the *single throughput* in the subsequent calculation of the channel occupancy time. Through curve fitting, we derived the following *sigmoid function* equation:

$$Thr. = \frac{a}{1 + e^{-\left(\frac{(120+Rss)-b}{c}\right)}}, \tag{3}$$

where $Thr.$ represents the estimated throughput (Mbps), and Rss is the received signal strength ($-dBm$) at the $Host_j$ position. The parameters a, b, and c represent the constants obtained from the parameter fitting with real-world measurement results. The parameter values for a, b, c, α, and W_K in the throughput estimation model will be optimized by the *parameter optimization tool*.

3.1.3. Throughput Reduction Factor

To account for the decrease in throughput caused by interference among hosts connected to the same AP, the concept of a *throughput reduction factor* was introduced. This factor enhances the precision of the *concurrent throughput* estimation under simultaneous communication [34]. The equation is as follows:

$$Thr._{con.} = Thr. \times srf(m), \tag{4}$$

where $Thr._{con.}$ represents the concurrent throughput of the host $Host_j$, $Thr.$ is the estimated single throughput between AP_i and $Host_j$, $srf(m)$ is the *throughput reduction factor*, and m is the number of hosts associated with the AP. Additionally, $srf(m)$ was empirically derived as the contention factor, which is given as follows:

$$srf(m) = \left(\frac{1}{m + \frac{0.1(m-1)}{4}} \right) \times 1 - (0.1 \times m - 1). \tag{5}$$

3.1.4. Parameter Optimization

The throughput estimation model relies on several parameters that significantly affect the accuracy of the estimation results. To optimize these parameters, we employ a parameter optimization tool that utilizes a *local search method*. This method combines a *tabu table* with a *hill-climbing* function to effectively prevent convergence to a local minimum [35].

To better demonstrate the improvements of our enhanced algorithm compared to previous studies, we utilized the same conditions as described in the previous paper. Consequently, the parameters of the throughput estimation model are consistent with those in Ref. [11]. The experimental scenarios and the parameters of the throughput estimation model will be introduced in Section 5.3.

3.2. Active AP Configuration Algorithm

The *active AP configuration algorithm* finds the optimal selection of the active APs and their host associations. The objective of the algorithm is to minimize the number of active APs that ensure that the *minimum host throughput constraint* is satisfied. To further reduce it, each AP is assumed to be equipped with dual interfaces.

3.2.1. Formulation

The formulation of the previous AP activation optimization problem is given as follows [11]:

1. **Inputs:**
   - APs' information (position, quantities);
   - Hosts' information (position, quantities);
   - Estimated single throughput for each AP_i and $Host_j$ pair: Tp_{ij};
   - Minimum throughput for the association: S;
   - Number of orthogonal channels (OCs) for each interface: C;
   - Minimum host throughput: G;
   - Available total throughput: B^a.
2. **Outputs:**
   - A collection of active APs equipped with dual interfaces;
   - A group of hosts connected to each interface at every active AP;
   - The channel assigned to each interface at every active AP.
3. **Objectives:**
   - E_1 denotes the count of active access points (APs) equipped with dual interfaces that must be minimized while adhering to the minimum host throughput constraint:

     $$E_1 = [activated\ APs'\ number]. \tag{6}$$

   - Adhering to the first objective, maximize the *minimum average host throughput* E_2:

     $$E_2 = min_j[Thr._{avg}], \tag{7}$$

     where $Thr._{avg}$ represents the average host throughput for AP_j that is given by:

     $$Thr._{avgj} = \frac{1}{\sum_k \frac{1}{Thr._{con.}}}, \tag{8}$$

where $Thr_{con.}$ represents the link speed between $node_j$ and $node_k$ ($link_{jk}$), calculated by Equation (5).

- Adhering to the two objectives, minimize the *total interfered communication time* E_3 for channel assignments:

$$E_3 = \sum_{i=1}^{N} \left[\sum_{k \in I_i, c_k = c_i} T_k \right], \quad (9)$$

where T_k represents the total communication time of AP_i, I_i represents the interference from APs at AP_i, and c_i is the channel assigned to AP_i.

4. **Constraints:**
   - Minimum host throughput: Each host must achieve an average throughput of at least G when all hosts are communicating simultaneously.
   - Total throughput: The combined throughput of all hosts must not exceed the available total throughput B^a.
   - Channel assignment: Every interface of an AP must be allocated a channel.

3.2.2. Algorithm Procedure

The *active AP configuration algorithm* is divided into the following three steps, and the pseudocode can be found in Appendix A:

1. **First Step**: In this initial phase, the algorithm identifies the active APs equipped with dual interfaces and determines their host connections. The objective is to reduce E_1 while enhancing E_2 [36].
   (1) **Preprocessing**: The algorithm begins with the input of AP and host locations. AP locations are manually selected within the network, considering factors such as electrical power supply, coverage, and user demands. The throughput for every possible AP/host pair is then estimated using the throughput estimation model outlined in Equation (3). Additionally, the *802.11n* interface of an AP is initially selected as the candidate interface for any host.
   (2) **Initial Solution Generation**: A *greedy algorithm* is used to calculate the initial solution E_1 [37].
   (3) **Host Association Improvement**:
   - **Host Reassociation for Maximum Throughput**: Reassign each host to the interface of the AP that provides the highest throughput, as determined by Equation (5), from among the available AP interfaces. Compute the cost function E_2 at this stage and set it as the best-found cost function, E_2^{best}.
   - **Identify Lowest Throughput Interface**: Find the interface of the AP that offers the lowest throughput to its host using Equation (7). Create a list of *modifiable hosts* associated with this interface that can connect to other AP interfaces.
   - **Random Reassociation of Modifiable Hosts**: Select one host at random from the *modifiable hosts list* and reassign it to a different active AP interface at random. Compute the new cost function E_2^{new}.
   - **Update Best Cost Function**: If E_2^{new} is greater than E_2^{best}, replace E_2^{best} with it and keep the new AP–host association. If not, revert to the previous association and maintain E_2^{best}.

   (4) **AP Selection Optimization**: This phase aims to optimize the number of active dual-interface APs and the associations between APs and hosts. The goal is to reduce both E_1 and E_2 metrics further using the *local search method* as described in [38].
   (5) **Link Speed Normalization**: The fairness criterion is applied if the total expected bandwidth exceeds B^a. Subsequently, the link speed is normalized.
   (6) **Termination Check**: For each active AP, if either of its two interfaces is found to be inactive, the interface should be activated, followed by executing the

host association improvement phase. The algorithm will move to the second phase if the minimum throughput requirement for the host is fulfilled. If this requirement is not satisfied, the algorithm will then proceed to the *AP selection optimization phase*.

2. **Second Phase**: In the second phase, a channel is assigned to each active AP interface to minimize E_3.
   (1) **Preprocessing**: Illustrate the network's interference and delay conditions using a graphical representation.
   (2) **Interfered AP Set Generation**: Identify the set of interfering AP interfaces for each AP interface.
   (3) **Initial Solution Construction**: Utilize a *greedy algorithm* to determine the initial solution.
   (4) **Solution Enhancement via Simulated Annealing**: Employ the probabilistic optimization method, *Simulated Annealing (SA)*, to progressively refine solutions. In this approach, SA is applied to optimize the channel assignment for each interface of every active AP, thereby improving network performance. The SA process is conducted at a fixed temperature T^{SA} for a predetermined number of iterations R^{SA}, with both T^{SA} and R^{SA} specified as algorithm parameters.
3. **Third Phase**: The third phase balances the loads across different channels to minimize E_3.
   (1) **Initialization**: Set all AP flags to 0 (*OFF*). This flag is used to ensure that each AP is processed only once.
   (2) **AP Selection**: Choose an AP currently marked as *OFF* and reassign its connected host to a different AP that uses another channel.
   (3) **Host Selection**: From the chosen AP, select one connected host for the AP reassignment process.
   (4) **Application of Change**: Finally, assign the host to a new AP.

3.3. Throughput Request Satisfaction Method

To address the issue of throughput unfairness and insufficiency among multiple hosts communicating simultaneously in a WLAN, the throughput request satisfaction method has been studied [13,14]. This approach employs three different types of throughput:

- Single Throughput: The *single throughput* S_i is determined when the corresponding host is the sole device communicating with the AP. This essentially reflects the *maximum throughput* achievable by the host in the absence of interference from other WLAN hosts.
- Concurrent Throughput: The *concurrent throughput* C_i is assessed when all hosts are communicating with their respective APs simultaneously within the WLAN. This measurement indicates the actual throughput of the host when subject to interference from other WLAN hosts.
- Target Throughput: The *target throughput* t_i for each host is computed on the basis that the total *channel occupancy time*, or cycle length, remains constant, even when the *concurrent throughput* is substituted with the *target throughput*.

The single throughput and the concurrent throughput are derived from measurements. Then, the *target throughput* is obtained from them. Finally, the *target throughput* is set as the *data rate* in traffic shaping with the PI control.

3.3.1. Channel Occupancy Time

To determine the appropriate target throughput for each host, the channel occupancy time is calculated based on the measured single throughput and concurrent throughput.

For the i-th host H_i, the channel occupancy time can be estimated by the ratio $\frac{C_i}{S_i}$. When all hosts communicate simultaneously, each host's channel occupancy time can be represented as $\frac{C_2}{S_2}$, ..., $\frac{C_n}{S_n}$, and their sum will be a constant for the data transmission cycle. If C_i is replaced with t_i, the total remains constant. Therefore, we derive the following equation:

$$\frac{C_1}{S_1} + \frac{C_2}{S_2} + \cdots + \frac{C_n}{S_n} = \frac{t_1}{S_1} + \frac{t_2}{S_2} + \cdots + \frac{t_n}{S_n}, \tag{10}$$

where S_i represents the *single throughput*, C_i represents the *concurrent throughput*, and t_i is the target throughput we demand.

3.3.2. Target Throughput for Fairness Allocation

In the *fairness throughput* allocation scenario, all the communicating hosts should be assigned the equal *target throughput*. Thus, the *fairness target throughput* F_i for host H_i satisfies: $F_1 = F_2 = F_3 = \cdots = F_n$. To transmit $F_1, F_2, F_3, \ldots, F_n$ (Mbit) data through the $S_1, S_2, S_3, \ldots, S_n$ link, the channel occupancy time can be calculated as follows:

$$\frac{C_1}{S_1} + \frac{C_2}{S_2} + \cdots + \frac{C_n}{S_n} = \frac{F_1}{F_1} + \frac{F_2}{S_2} + \cdots + \frac{F_n}{S_n},$$

$$F_1 = F_2 = \cdots = F_n = \frac{\sum_{i=1}^{n} \frac{C_i}{S_i}}{\sum_{i=1}^{n} \frac{1}{S_i}},$$

where S represents the single throughput, C is the concurrent throughput, and F is the calculated fairness target throughput for each host.

3.3.3. Traffic Shaping

To realize the control of actual throughput, we deployed the *traffic shaping* method. *Traffic shaping* manages network bandwidth through the scheduling, policing, shaping, and classification of traffic. In Linux, this can be achieved using the *tc* command, which includes queuing discipline (*qdisc*), classes, and filters [39].

We utilized the *classful HTB (Hierarchical Token Bucket) qdisc* to regulate traffic at a specified data rate, d_i. The HTB employs *token buckets* to distribute traffic across different classes, governed by two parameters: *ceil* and *data rate*. These parameters define the allocated and maximum bandwidth, respectively. In this study, we set both parameters to identical values to maintain the desired quality of service across various traffic classes.

3.3.4. PI Controller of Rate and Ceil Parameters

In the field of *traffic shaping*, the *tc command* controlling the data rate parameter d_i can only determine the maximum upper limit for a host's traffic. However, it cannot always guarantee that the actual throughput satisfies the target throughput. To this end, the *PI (Proportional-Integral) feedback control* mechanism is utilized. For each time step m (60 s in this paper), by calculating the error space $t_i - R_i(m)$ between the measured actual throughput and the target, under the proper adjustment of the proportional gain and the integral gain, the size of the input *data rate* d_i of the system is effectively selected, so that the actual throughput is as close as possible to the target. The equation is as follows:

$$d_i(m) = d_i(m-1) + K_P \times (R_i(m-1) - R_i(m)) + K_I \times (t_i - R_i(m)),$$

where $R_i(m)$ represents the actual throughput result at each time step m, and K_P and K_I represent the parameters of *P-control* gain and *I-control* gain, respectively. In this paper, $K_P = 0.3$ and $K_I = 0.7$ are used, which have been experimentally adjusted in real-world situations where they can quickly and accurately control d_i to meet the desired target.

3.4. Limitations of the Active AP Configuration Algorithm

In the previous *active AP configuration algorithm*, only the average throughput among the hosts associated with the same AP can satisfy the minimum host throughput constraint. If there is a host located far from the AP, this host may not satisfy this constraint, since the throughput difference from other hosts located near the AP can be large.

This problem must be solved in this paper by introducing the *throughput request satisfaction method* to these hosts. With this method, the *target throughput* is introduced to them. If it does not satisfy the given initial minimum host throughput G for the constraint (5 Mbps in evaluations of this paper), the active AP configuration is reconstructed by applying the algorithm with the increased tentative minimum host throughput that is introduced to increase the number of active APs. Then, the *target throughput* is recalculated and checked in the enhanced active AP configuration algorithm in this paper, which will be presented in the next section.

4. Enhanced Active AP Configuration Algorithm

In this section, we present the *enhanced active AP configuration algorithm* by introducing the *throughput request satisfaction method*. Each AP is equipped with dual interfaces for minimizing active APs.

4.1. Enhanced Active AP Configuration Algorithm Procedure

Figure 1 illustrates the flowchart of the proposed enhanced active AP configuration algorithm. It outlines the application process of the proposed algorithm for fairness host throughput in an energy-efficient WLAN environment. And the Appendix B shows the pseudocode of our enhanced algorithm.

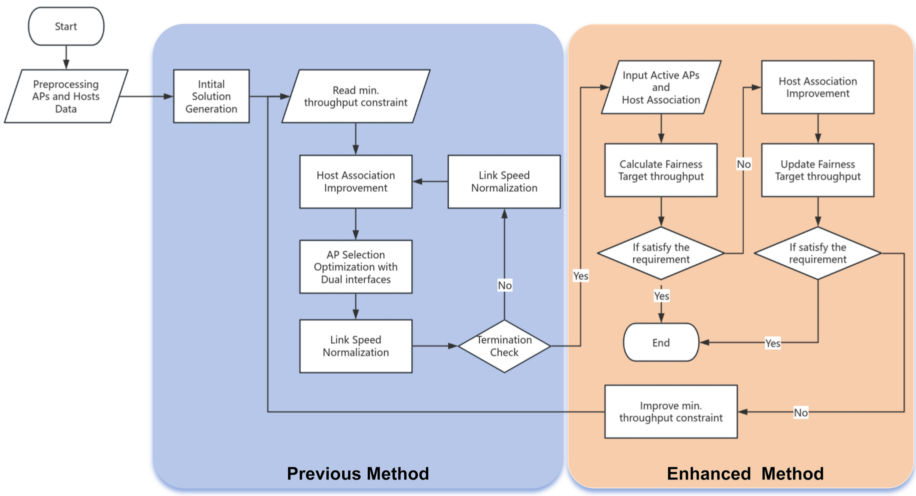


Figure 1. Flow of enhanced active AP configuration algorithm.

1. Set the network field layout, including the locations of APs and hosts, as well as the walls or obstacles, and initialize the tentative minimum host throughput $G = 5$ Mbps in the problem.
2. Apply the *AP active configuration algorithm* to the network field layout to find the *active AP configuration* including the active APs, their channels, and associated hosts with the tentative minimum host throughput G.
3. Apply the *throughput request satisfaction method* to the obtained *active AP configuration* to calculate the fair *target throughput* for the hosts.
   (1) Calculate the *single throughput* S_i for each AP–host pair using the *throughput estimation model*.
   (2) Calculate the *concurrent throughput* C_i from the *single throughput* with the *throughput reduction factor*.
   (3) Calculate the fair *target throughput* from them.
   (4) Terminate the procedure if the *target throughput* is equal to or larger than G. Otherwise, go to Step (5).

(5) Increase G by the *throughput constraint update* in the following subsection and go to Step 2.

4. Apply *traffic shaping* to the hosts at the AP to control the throughput at the *target throughput*, while adjusting the *data rate* parameter d_i by the *PI control*, and measure the actual throughput of all the hosts.

4.2. Throughput Constraint Update

If the hosts' association and activated APs given by the active AP configuration cannot satisfy the current minimum host throughput constraint, in order to increase the number of activated APs, the tentative minimum host throughput G_{new} is calculated with the following equation:

$$G_{new} = G + \Delta G, \tag{11}$$

where ΔG represents the given throughput increase unit (1 Mbps in this paper).

5. Evaluation

In this section, we evaluate the proposal through simulations using the *WIMNET simulator* [16] and experiments using the testbed system.

5.1. Evaluation Setup

Here, we introduce the setup for the simulation environment and experimental environment.

5.1.1. Simulation

The *WIMNET simulator* is employed in this paper for simulation purposes. It was designed to assess the performance of large-scale wireless Internet access mesh networks on a standard PC within a reasonable CPU time. In this study, it has been used to simulate various WLAN configurations, including different topologies, channel models, and interference conditions. Tables 1 and 2 outline the hardware and software specifications utilized in the simulations.

Table 1. PC environment.

WIMNET Simulator PC	Configuration
CPU	Intel Core i7
Memory	8 GB
OS	Ubuntu LTS 14.04

Table 2. Simulation parameters in the WIMNET simulator.

Parameter	Value
packet size	1500 bytes
max. transmission rate	150 Mbit/s
propagation model	log distance path loss model
rate adaptation algorithm	link speed estimation model
carrier sense threshold	85 dBm
transmission power	19 dBm
collision threshold	10
RTS/CTS	yes

5.1.2. Experimental Setup

Table 3 shows the hardware and software configurations of the testbed system. *Raspberry Pi 4B* is used for each AP by running *hostapd* [40]. The embedded wireless NIC is utilized for *interface 1*, with the 2.4 GHz *802.11n* protocol of dual interfaces, while the *Archer T4U wireless NIC* [41] adapter is used for *interface 2*, with the 5 GHz *802.11ac* protocol.

The 40 MHz bonded channel is used at both interfaces. *Linux*-based laptop PCs are used for the client hosts and the server. And we used the software *iperf* [42] to test the throughput.

The testbed system using *Raspberry Pi* APs with single interfaces was previously implemented and used in our studies [11,36]. In this testbed system, *Archer T4U* is adopted at the AP to enable dual-interface functions, as illustrated in Figure 2.

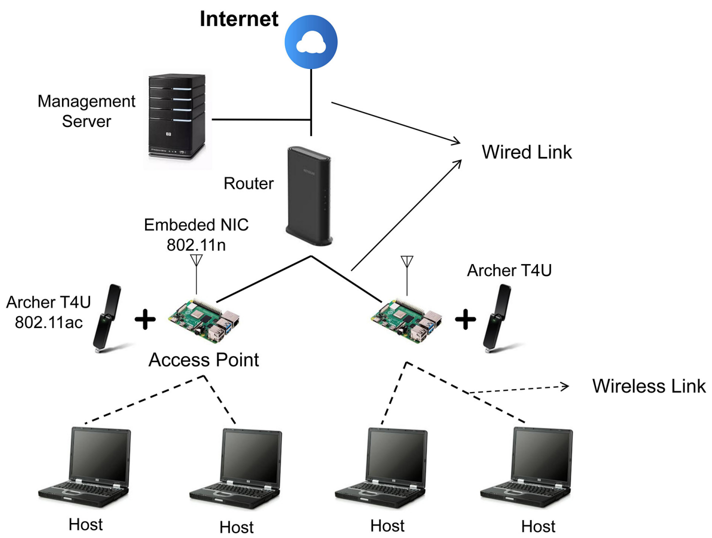


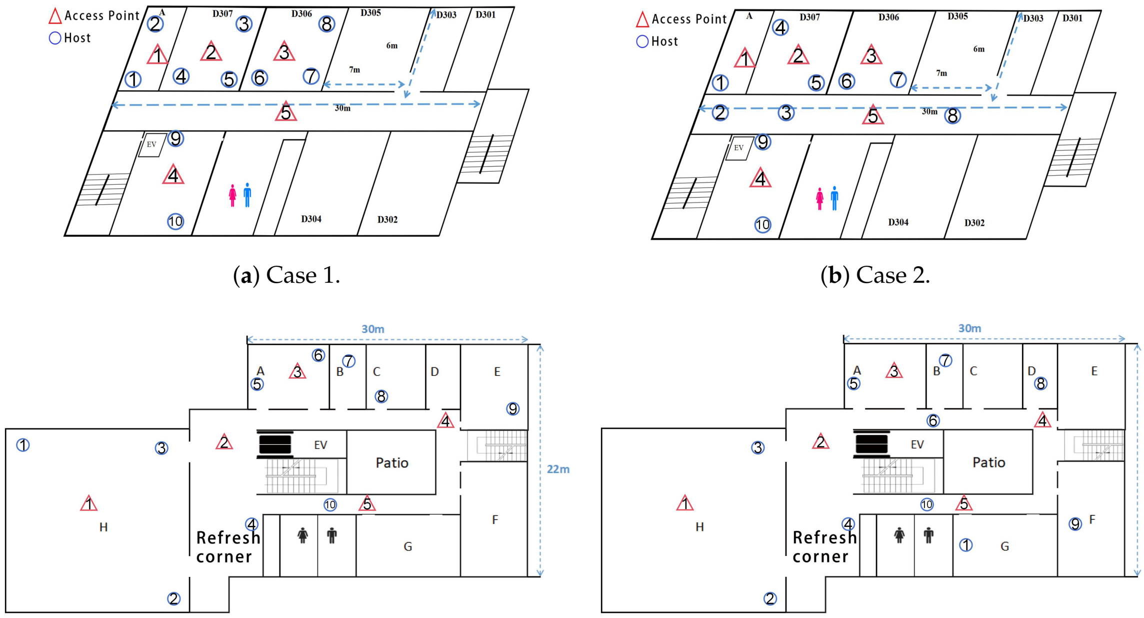
Figure 2. Topology of the testbed system.

Table 3. Hardware and software of the testbed system.

Host PC	
type	1. Toshiba Dynabook R731/B 2. Toshiba Dynabook R734/K 3. Fujitsu Lifebook S761/C
OS	Linux Ubuntu 14.04 LTS (kernel 3.13.0-57)
CPU	1. Intel Core i5-2520M @2.5 GHz 2. Intel Core i5-4300M @2.6 GHz 3. Intel Core i5-2520M @2.5 GHz
RAM	4GB DDR3-1333 MHz
software	*iperf 2.0.5*
Server PC	
type	Fujitsu Lifebook S761/C
CPU	Intel Core i5-2520M @2.5 GHz
RAM	4 GB DDR3 1333 MHz
OS	Linux Ubuntu 14.04 LTS (kernel 3.13.0-57)
software	*iperf 2.0.5*
Access Point	
type	Raspberry Pi 4B
OS	Linux (Raspbian)
CPU	Broadcom BCM2711 @1.5 GHz
RAM	8 GB LPDDR4-3200 SDRAM
NIC	BCM4345/6
external NIC	Archer T4U V3.0 AC1300
software	*hostapd v2.10*

5.2. Network Fields and Cases for Device Locations

Figure 3 shows the four cases of network topologies in two network fields for experiments, namely, Engineering Building #2 and the Graduate School of Natural Sciences Building at Okayama University, Japan. In each case, five dual-interface APs and ten hosts are placed at various locations considering the signal coverage in the field. The hosts are randomly placed to investigate various host positions in real-world scenarios.



(**a**) Case 1. (**b**) Case 2.

(**c**) Case 3. (**d**) Case 4.

Figure 3. Device locations in network fields. The red triangle in the figure indicates the location of the AP (Access Point), while the blue circle denotes the host location. Case 1 and Case 2 refer to locations in the Engineering Building #2 at Okayama University, and Case 3 and Case 4 refer to the Graduate School of Natural Sciences Building at Okayama University.

5.3. Throughput Estimation Model Setup

To estimate the *single throughput*, the *concurrent throughput*, and the *fairness target throughput*, a *Python* program for the *throughput estimation model* was implemented. Table 4 shows the parameter values of the model that were optimized for the network fields in Figure 3.

Table 4. Parameters for the throughput estimation model.

Parameter	Field (a) Engineering Building #2		Field (b) Graduate School Building	
	802.11*n*	**802.11*ac***	**802.11*n***	**802.11*ac***
P_1	−28.9	−31.0	−28.5	−30.5
α	2.2	2.15	1.7	2.0
W_1	7.21	2.1	6.5	2.3
W_2	6.9	8.5	4.2	6.4
W_3	3.4	3.7	3.1	1.8
W_4	4.7	1.8	1.5	4.2
W_5	2.11	7.0	2.0	4.3
W_6	2.5	1.5	2.0	5.3
a	63.5	133	65.0	134.5
b	62.0	58.0	62.0	58.5
c	6.78	6.30	6.78	6.25

5.4. Results and Discussions

Here, we first discuss simulation results using the *WIMNET simulator* in the four cases. Based on these simulation results, we conducted experiments to verify the effectiveness of our proposal in real-world applications.

5.4.1. Case 1

First, we discuss simulation results in *Case 1*. The given initial *minimum host throughput* G is set to 5 Mbps. The activated APs and the host associations are given by the *enhanced active AP configuration algorithm* before applying the *throughput constraint update* as follows:

- AP_{2_1}: H_2, H_5, H_7;
- AP_{2_2}: H_1, H_3, H_4, H_6, H_8, H_9, H_{10},

where AP_2 represents AP_2 in the location map, and _1 and _2 represent the interface for 2.4 GHz and the interface for 5 GHz, respectively. Table 5 shows the simulation results.

Table 5. Simulation results before *throughput constraint update* in *Case 1*.

Throughput (Mbps)	AP_{2_1}			AP_{2_2}						
	H_2	H_5	H_7	H_1	H_3	H_4	H_6	H_8	H_9	H_{10}
S	38.28	55.26	30.46	88.77	128.2	127.2	99.94	71.94	112.9	53.34
C	10.04	14.49	7.99	4.97	7.17	7.12	5.59	4.02	6.32	2.98
F	10.21	10.21	10.21	4.99	4.99	4.99	4.99	4.99	4.99	4.99

S represents the single throughput, C represents the concurrent throughput, and F represents the fair target throughput. The data highlighted in red indicates results that fail to meet the current throughput constraints after the throughput fairness calculation.

In this table, S represents the *single throughput* and C is the *concurrent throughput* that are obtained by the previous algorithm, while F represents the fair *target throughput* found by applying the *fair throughput request satisfaction method*. Unfortunately, in Table 5, the fair *target throughput* calculation for the seven hosts associated with AP_{2_2} does not satisfy G (=5 Mbps). Then, the *throughput constraint update* is applied to satisfy the fairness throughput allocation by increasing it to G_{new}. The activated APs and the host associations are given by the algorithm as follows:

- G_{new}: 6 Mbps;
- AP_{2_1}: H_2, H_3, H_5, H_7;
- AP_{2_2}: H_1, H_4, H_6, H_8, H_9, H_{10}.

Table 6 shows the estimated throughput results. This time, any calculated fair *target throughput* result satisfies the current G_{new} (6 Mbps).

Table 6. Simulation results after the *throughput constraint update* in *Case 1*.

Throughput (Mbps)	AP_{2_1}				AP_{2_2}					
	H_2	H_3	H_5	H_7	H_1	H_4	H_6	H_8	H_9	H_{10}
S	38.28	55.26	55.9	30.46	88.77	127.2	99.94	71.94	112.9	53.34
C	6.58	9.6	9.49	5.23	7.25	10.39	8.16	5.87	9.22	4.35
F	7.24	7.24	7.24	7.24	6.93	6.93	6.93	6.93	6.93	6.93

Subsequently, the final host associations and activated APs were input into our experiment. In Figure 4, the red column represents the concurrent throughput result obtained using the previous algorithm. Although it can reduce the number of activated APs and save energy, it fails to provide the throughput G_{new} (6 Mbps) in real-world applications for some hosts ($H_2, H_6, H_7, H_9, H_{10}$). By comparing the data presented in Figure 4, it is evident that, following our proposal, each host achieved an actual throughput exceeding G_{new}, and the fairness among hosts is significantly enhanced.

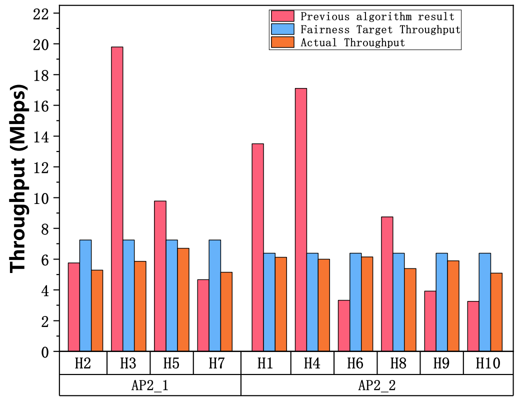


Figure 4. Experimental results for *Case 1*.

5.4.2. Case 2

Second, we discuss results in *Case 2*. For simulation, the activated APs and the host associations are given by the algorithm before the *throughput constraint update* as follows, where Table 7 shows the simulation results:

- AP_{5_1}: H_2, H_7, H_8, H_9;
- AP_{5_2}: H_1, H_3, H_4, H_5, H_6, H_{10}.

In Table 7, the fair *target throughput* results for the four hosts associated with AP_{2_1} do not satisfy the initial G (5 Mbps). Although the average throughput can achieve 5 Mbps, the result of the fairness throughput allocation shows that the hosts connected to AP_{2_1} suffer from insufficient throughput.

Table 7. Simulation results before the *throughput constraint update* in *Case 2*.

Throughput (Mbps)	AP_{2_1}				AP_{2_2}					
	H_2	H_7	H_8	H_9	H_1	H_3	H_4	H_5	H_6	H_{10}
S	8.44	27.08	14.9	56.0	88.77	127.2	99.94	71.94	112.9	53.34
C	6.58	9.6	9.49	5.23	7.25	10.39	8.16	5.87	9.22	4.35
F	2.86	2.86	2.86	2.86	7.15	7.15	7.15	7.15	7.15	7.15

The data highlighted in red indicates results that fail to meet the current throughput constraints after the throughput fairness calculation.

Then, the *throughput constraint update* is repeatedly applied, and the *target throughput* G_{new} = 12 Mbps can be satisfied after gradual increases. Table 8 shows the improved estimated throughput results where any fair *target throughput* result satisfies the minimum throughput constraint. The activated APs and the host associations are given by the algorithm as follows:

- G_{new}: 12 Mbps;
- AP_{3_1}: H_9;
- AP_{3_2}: H_4, H_5, H_8;
- AP_{4_1}: H_3, H_7;
- AP_{4_2}: H_1, H_2, H_6, H_{10}.

Table 8. Simulation results after the *throughput constraint update* in *Case 2*.

Throughput (Mbps)	AP_{3_1}	AP_{3_2}			AP_{4_1}		AP_{4_2}			
	H_9	H_4	H_5	H_8	H_3	H_7	H_1	H_2	H_6	H_{10}
S	21.26	70.5	102.1	95.72	33.88	48.38	87.59	93.09	124.9	114.1
C	21.26	18.49	26.79	25.11	15.06	21.5	15.05	15.99	21.46	19.61
F	21.26	22.86	22.86	22.86	17.71	17.71	17.65	17.65	17.65	17.65

For the updated minimum throughput constraint $G_{new} = 12$ Mbps, the simulation result verify that all hosts can enjoy more than this constraint; meanwhile, after the experiment using our proposal, the actual throughput shown in Figure 5 has been balanced with fairer allocation and can achieve the updated G_{new}.

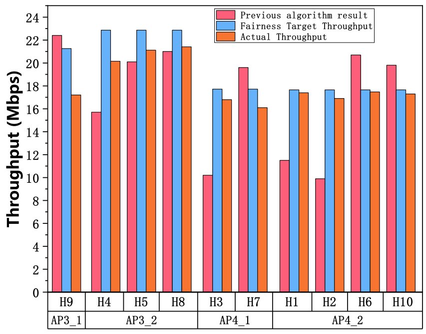


Figure 5. Experimental results for *Case 2*.

5.4.3. Case 3

Third, we discussed simulation and experiment results in *Case 3*. The activated APs and the host associations are given by the algorithm before the *throughput constraint update* as follows, where Table 9 shows the simulation results:

- AP_{2_1}: H_8, H_9;
- AP_{2_2}: H_1, H_2, H_3, H_4, H_5, H_6, H_7, H_{10}.

Table 9. Simulation results before the *throughput constraint update* in *Case 3*.

Throughput (Mbps)	AP_{2_1}		AP_{2_2}							
	H_8	H_9	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_{10}
S	43.9	27.83	56.32	66.16	123.54	114.2	124.8	84.34	81.02	65.79
C	19.51	12.37	2.07	2.43	4.53	4.19	4.58	3.1	2.97	2.41
F	15.14	15.14	3.02	3.02	3.02	3.02	3.02	3.02	3.02	3.02

The data highlighted in red indicates results that fail to meet the current throughput constraints after the throughput fairness calculation.

In Table 9, the fair *target throughput* results for the eight hosts associated with AP_{2_2} do not satisfy the initial G. Then, the *throughput constraint update* is applied, where G_{new} = 6 Mbps can satisfy G after it is gradually increased. The activated APs and the host associations are given by the algorithm as follows:

- G_{new}: 6 Mbps;
- AP_{2_1}: H_3, H_5, H_8, H_9;
- AP_{2_2}: H_1, H_2, H_4, H_6, H_7, H_{10}.

Table 10 shows the estimated throughput results where any fair *target throughput* result satisfies G. The experiment results are shown in Figure 6.

Table 10. Simulation results after the *throughput constraint update* in *Case 3*.

Throughput (Mbps)	AP_{2_1}				AP_{2_2}					
	H_3	H_5	H_8	H_9	H_1	H_2	H_4	H_6	H_7	H_{10}
S	59.41	59.9	43.9	27.83	56.32	66.16	114.19	84.34	81.02	65.79
C	10.21	10.21	10.11	4.78	4.6	5.4	9.32	6.88	6.61	5.37
F	7.45	7.45	7.45	7.45	6.04	6.04	6.04	6.04	6.04	6.04

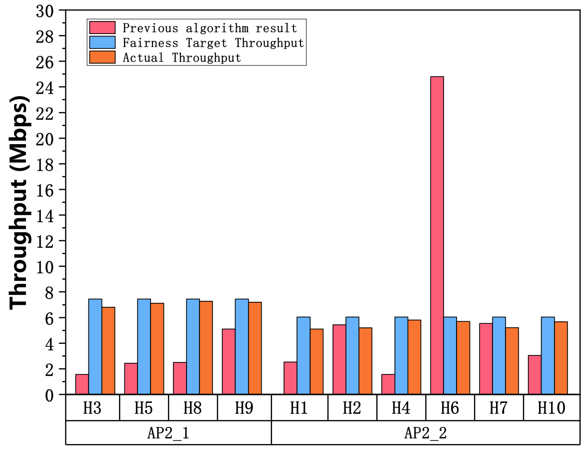


Figure 6. Experimental results for *Case 3*.

In Figure 6, it can be observed that there is a significant difference in concurrent throughput with the previous algorithm. This discrepancy is due to the variations in RSS caused by the distance between the host and the AP, as well as environmental interferences that can impact the actual values. With our algorithm, it is evident that the new experimental results have significantly improved this issue. Each host can now meet the minimum throughput constraint.

5.4.4. Case 4

Fourth, we discuss the results in *Case 4*. The activated APs and the host associations are given by the algorithm before the *throughput constraint update* as follows, where Table 11 shows the simulation results:

- AP_{3_1}: H_1, H_7;
- AP_{3_2}: H_2, H_3, H_4, H_5, H_6, H_8, H_9, H_{10}.

Table 11. Simulation results before the *throughput constraint update* in *Case 4*.

Throughput (Mbps)	AP_{3_1}		AP_{3_2}							
	H_1	H_7	H_2	H_3	H_4	H_5	H_6	H_8	H_9	H_{10}
S	30.17	59.51	25.22	72.66	70.86	118.0	125.7	100.9	24.27	62.7
C	13.41	26.45	0.93	2.67	2.60	4.33	4.61	3.71	0.89	2.30
F	17.8	17.8	1.95	1.95	1.95	1.95	1.95	1.95	1.95	1.95

The data highlighted in red indicates results that fail to meet the current throughput constraints after the throughput fairness calculation.

In Table 11, the fair *target throughput* results for the eight hosts associated with AP_{3_2} do not satisfy G. Then, the *throughput constraint update* is repeatedly applied, where $G_{new} = 8$ Mbps can satisfy G after it is gradually increased. The activated APs and the host associations are given by the algorithm as follows:

- G_{new}: 8 Mbps;
- AP_{3_1}: H_3, H_5;
- AP_{3_2}: H_2, H_4;
- AP_{4_1}: H_1, H_7;
- AP_{4_2}: H_6, H_8, H_9, H_{10}.

Table 12 shows the estimated throughput results where any fair *target throughput* result satisfies G. Figure 7 shows the effectiveness of our proposal in a real-world experiment.

Table 12. Simulation results after the *throughput constraint update* in *Case 4*.

Throughput (Mbps)	AP_{3_1}		AP_{3_2}		AP_{4_1}		AP_{4_2}			
	H_3	H_5	H_2	H_4	H_1	H_7	H_6	H_8	H_9	H_{10}
S	44.48	58.46	25.22	70.86	37.77	43.26	98.77	129.3	105.5	101.03
C	19.77	25.98	11.21	31.49	16.79	19.23	16.97	22.22	18.13	17.35
F	22.45	22.45	16.53	16.53	17.93	17.93	18.46	18.46	18.46	18.46

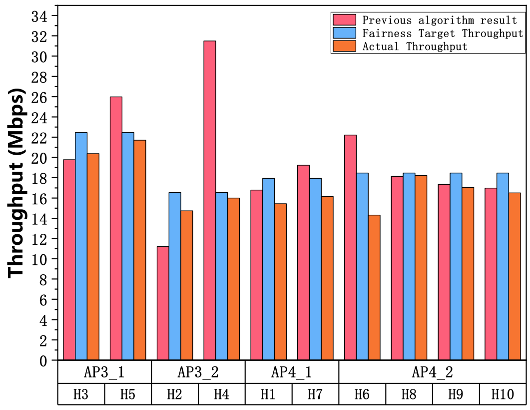


Figure 7. Experimental results for *Case 4*.

5.4.5. Fairness Comparison

The simulation and experiment results show that every host satisfied the *minimum host throughput* after applying the *throughput constraint update* of the *enhanced active AP configuration algorithm*. These results confirm the validity and effectiveness of the proposal. Figure 8 compares the throughput distribution between the previous algorithm and the enhanced algorithm across four cases.

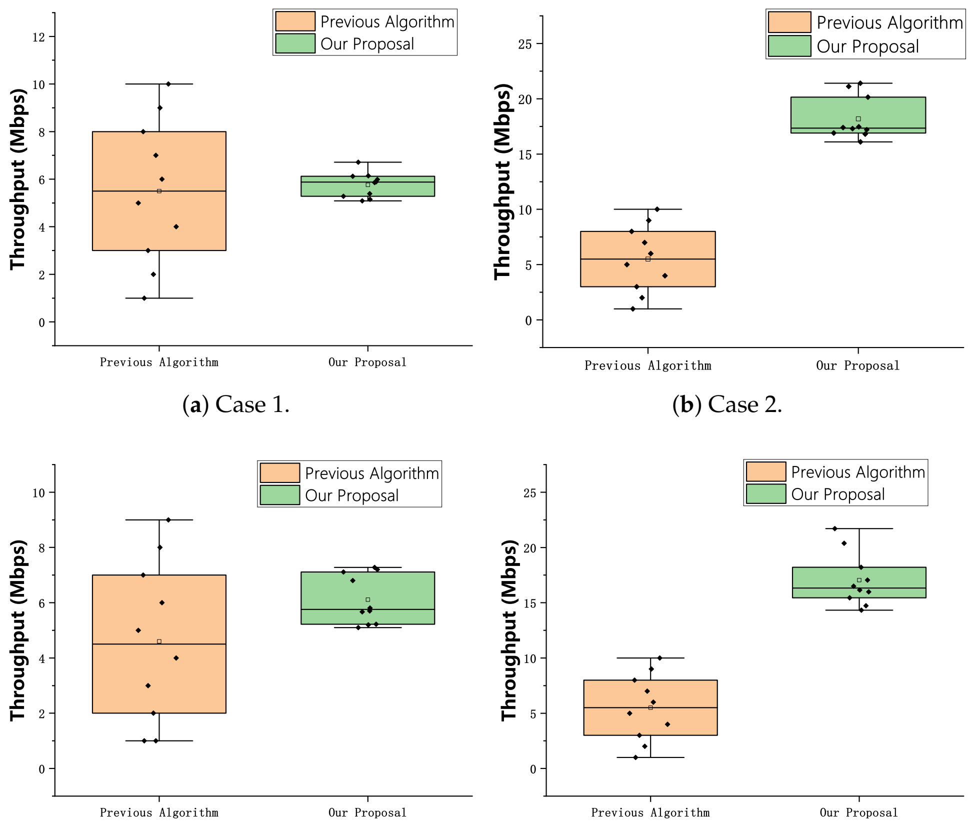
In *Case 1*, the activated access point (AP) remains the same as before, with only one active AP in the scenario. The median throughput results, shown in Figure 8a, highlight an increase in efficiency after implementing our improved active AP configuration algorithm. The data points in the box plot are densely clustered around the median and display an approximately symmetrical distribution, indicating minimal skewness. This pattern illustrates the algorithm's effectiveness in consistently and fairly distributing guaranteed throughput.

In *Case 2*, as shown in Figure 8b, our enhanced algorithm results in higher throughput performance. As mentioned in Section 5.4.2, the number of activated access points in the network only increases by one compared to the previous algorithm case. This adjustment still achieves significant energy savings compared to scenarios without any optimization measures. Additionally, our algorithm demonstrates a narrower range between the highest and lowest values, indicating reduced variability. The majority of the data points are closely clustered, supporting the goal of fair resource allocation.

In *Case 3*, as shown in Figure 8c, although the maximum network throughput experiences certain limitations, the box plot demonstrates that the throughput control method in our algorithm effectively regulates the actual traffic for each host, achieving a fairer distribution. This method ensures that even the minimum throughput in the actual test network meets the desired throughput constraints, thereby addressing the potential issue of the host's insufficient throughput in practical applications, which was a concern with the previous method.

In *Case 4*, the situation is similar to *Case 2*. Although the number of activated APs is increased to ensure that individual hosts can achieve throughput greater than the constraints, the increase is limited to just one AP. This reduces the overall consumption of APs in the network compared to when this measure is not applied. Additionally, by examining the data distribution, as well as the maximum, minimum, and median values in Figure 8c, it is evident that fairness has improved compared to the results obtained with the previous algorithm.

The discussion above fully demonstrates that our proposed algorithm, by adopting the *throughput request satisfaction method*, reduces the throughput variations among hosts. It not only maintains an average throughput comparable to the previous algorithm but also achieves a fairer distribution, preventing some hosts from having excessively high throughput while others have insufficient throughput.



(**a**) Case 1. (**b**) Case 2.

(**c**) Case 3. (**d**) Case 4.

Figure 8. Throughput distributions for the previous and enhanced algorithms.

6. Conclusions

In this paper, we presented the *enhanced active AP configuration algorithm* by incorporating the *throughput request satisfaction method* to control the *actual throughput* at the fair *target throughput* for every host by applying *traffic shaping* at the AP. This is an extension of the previous *active AP configuration algorithm* that addresses the issue of part of the host suffering from insufficient and unfair concurrent throughput.

To address this issue, we deployed dual-interface device support for higher access capacity and reduced the number of APs; in addition, the throughput control phase provided the actual throughput of each host. It calculates the *target throughput* from the *single* and *concurrent throughput* of each host. If it does not satisfy the required throughput, the tentative *minimum throughput* is increased by the *throughput constraint update*, and the *active AP configuration* and the *target throughput* are recalculated.

For evaluations, in four topology cases with five APs and 10 hosts, we conducted simulations using the *WIMNET simulator* and experiments using the *testbed system* with *Raspberry Pi 4B* for APs. The results show that the proposal always achieved the required *minimum throughput* in simulations and in experiments and, at the same time, the number of activated APs has obviously been reduced to only one or two. Thus, the validity and effectiveness of our proposal were confirmed. In future work, we will further enhance the algorithm by considering the transmission power control at the AP and evaluate it using different protocols such as *802.11ax* in various network scenarios.

Author Contributions: Conceptualization, B.W. and N.F.; methodology, B.W.; software, B.W.; data curation, B.W., D.K., X.W. and T.S.; writing—original draft preparation, B.W.; writing—review and editing, N.F. and Y.-C.F.; validation, B.W., D.K., X.W. and T.S.; supervision, N.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors thank the reviewers for their thorough reading and helpful comments.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Algorithm A1: Previous Active AP Configuration Algorithm

```
Result: AP configurations: APs, Host associations: HAs, Channel assignments: CAs
Function Initial_Solution(AP_Info, H_Info):
    Data: Access Point information (AP_Info), Host information (H_Info)
    Result: Initial Solution E_1 based on Greedy algorithm
return E_1
Function Host_Association_Improvement():
    AP_min ← FindAPWithLowestThroughput(APs);
    ModifiableHosts ← GetModifiableHosts(AP_min, APs);
    // Randomly reassociate one modifiable host
    Host ← SelectRandom(ModifiableHosts);
    Reassociate Host to a different AP;
    Calculate E_2^new for new configuration;
    if E_2^new ≥ E_2^best then
        E_2^best ← E_2^new;
    end
    else
        Rollback;
    end
return Improved Association
Function Optimize_AP_Selection(APs, Associations):
    // Local search to optimize AP selection and host associations
    E_1, E_2 ← LocalSearch(); // Perform the local search
return APs, Associations
Function MainFunction():
    Initial_Solution();
    Host_Association_Improvement();
    // Termination Check
    while True do
        foreach AP in APs do
            if either interface of AP is not activated then
                ActivateInterface(AP);
                Host_Association_Improvement();
                if minimum throughput constraint is satisfied then
                    assigns a channel to each active AP interface to minimize E_3;
                end
                else
                    Optimize_AP_Selection(APs, Hosts);
                end
            end
        end
        if minimum throughput constraint is satisfied for all APs then
            Break;
        end
        else
            Host_Association_Improvement();
        end
    end
    Channel_Load_Averaging();
return
```

Appendix B

Algorithm A2: Enhanced Active AP Configuration Algorithm for Fair Throughput

```
Data: Network layout including APs and hosts, throughput requirements
Result: Optimized AP configurations, host associations, and fair throughput allocations
Function EnhancedActiveAPConfiguration():
    // Initialize network and minimum throughput
    Initialize network with APs and hosts;
    Set tentative minimum host throughput G = 5 Mbps;
    repeat
        // Apply previous active AP configuration algorithm
        Apply_Active_AP_Config_Algorithm();
        // Apply throughput request satisfaction method
        foreach host in Hosts do
            S_i ← CalculateSingleThroughput(host);
            C_i ← CalculateConcurrentThroughput(host);
            T_i ← CalculateTargetThroughput(S_i, C_i);
        end
        if all T_i ≥ G then
            return AP configurations, host associations;
        end
        else
            // Increase G if minimum throughput not met
            G ← G + ΔG;
        end
    until throughput requirements are met;
    // Apply traffic shaping to adjust data rates
    ApplyTrafficShaping();
    MeasureActualThroughputs();
return
```

References

1. Ersoy, M.; Yiğit, T.; Yüksel, A.S. A decision support tool for indoor 801.11 ac wlan modeling using optimization techniques. *El-Cezeri J. Sci. Eng.* **2020**, *7*, 1231–1244. [CrossRef]
2. Kassa, L.; Davis, M.; Cai, J.; Deng, J. A New Adaptive Frame Aggregation Method for Downlink WLAN MU-MIMO Channels. *J. Commun.* **2021**, *16*, 311–322. [CrossRef]
3. Oh, H.S.; Jeong, D.G.; Jeon, W.S. Joint radio resource management of channel-assignment and user-association for load balancing in dense WLAN environment. *IEEE Access* **2020**, *8*, 69615–69628. [CrossRef]
4. Crow, B.P.; Widjaja, I.; Kim, J.G.; Sakai, P.T. IEEE 802.11 wireless local area networks. *IEEE Commun. Mag.* **1997**, *35*, 116–126. [CrossRef]
5. Mahboob, T.; Lee, H.Y.; Shin, M.; Chung, M.Y. SDN-based centralized channel assignment scheme using clustering in dense WLAN environments. *Wirel. Pers. Commun.* **2020**, *114*, 2693–2716. [CrossRef]
6. Ali, R.; Nauman, A.; Zikria, Y.B.; Kim, B.S.; Kim, S.W. Performance optimization of QoS-supported dense WLANs using machine-learning-enabled enhanced distributed channel access (MEDCA) mechanism. *Neural Comput. Appl.* **2020**, *32*, 13107–13115. [CrossRef]
7. Pirayesh, H.; Sangdeh, P.K.; Zeng, H. Coexistence of wi-fi and iot communications in wlans. *IEEE Internet Things J.* **2020**, *7*, 7495–7505. [CrossRef]
8. Zhang, H.; Duan, Y.; Long, K.; Leung, V.C. Energy efficient resource allocation in terahertz downlink NOMA systems. *IEEE Trans. Commun.* **2020**, *69*, 1375–1384. [CrossRef]
9. Mukherjee, A.; Goswami, P.; Khan, M.A.; Manman, L.; Yang, L.; Pillai, P. Energy-efficient resource allocation strategy in massive IoT for industrial 6G applications. *IEEE Internet Things J.* **2020**, *8*, 5194–5201. [CrossRef]
10. Palacios, R.; Granelli, F.; Gajic, D.; Li, C.; Kliazovich, D. An energy-efficient point coordination function using bidirectional transmissions of fixed duration for infrastructure IEEE 802.11 WLANs. In Proceedings of the 2013 IEEE International Conference on Communications (ICC), Budapest, Hungary, 9–13 June 2013; pp. 2443–2448.
11. Roy, S.C.; Funabiki, N.; Rahman, M.M.; Wu, B.; Kuribayashi, M.; Kao, W.-C. A Study of the Active Access-Point Configuration Algorithm under Channel Bonding to Dual IEEE 802.11n and 11ac Interfaces in an Elastic WLAN System for IoT Applications. *Signals* **2023**, *4*, 274–296. [CrossRef]
12. Wu, B.; Funabiki, N.; Roy, S.C.; Rahman, M.M.; Kong, D.; Fang, S. An Application of Throughput Request Satisfaction Method for Maximizing Concurrent Throughput in WLAN for IoT Application System. *Sensors* **2024**, *24*, 2173. [CrossRef] [PubMed]
13. Rahman, M.M.; Funabiki, N.; Munene, K.I.; Roy, S.C.; Kuribayashi, M.; Gulo, M.M.; Kao, W.C. A Throughput Request Satisfaction Method for Concurrently Communicating Multiple Hosts in Wireless Local Area Network. *Sensors* **2022**, *22*, 8823. [CrossRef] [PubMed]

14. Rahman, M.M.; Funabiki, N.; Munene, K.I.; Roy, S.C.; Kuribayashi, M.; Kao, W.C. A Throughput Fairness Control Method for Concurrent Communications in Wireless Local-Area Network with Multiple Access-Points. *J. Commun.* **2022**, *17*, 592–599. [CrossRef]
15. Munene, K.I.; Funabiki, N.; Briantoro, H.; Rahman, M.M.; Akhter, F.; Kuribayashi, M.; Kao, W.C. A throughput drop estimation model for concurrent communications under partially overlapping channels without channel bonding and its application to channel assignment in IEEE 802.11 n WLAN. *IEICE Trans. Inf. Syst.* **2021**, *104*, 585–596. [CrossRef]
16. Funabiki, N. *Wireless Mesh Networks*; InTech-Open: London, UK, 2011. Available online: https://www.intechopen.com/books/26 (accessed on 10 June 2024).
17. Parashar, V.; Kashyap, R.; Rizwan, A.; Karras, D.A.; Altamirano, G.C.; Dixit, E.; Ahmadi, F. Aggregation-based dynamic channel bonding to maximise the performance of wireless local area networks (WLAN). *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 4464447. [CrossRef]
18. Lanante, L.; Roy, S. Analysis and optimization of channel bonding in dense IEEE 802.11 WLANs. *IEEE Trans. Wirel. Commun.* **2020**, *20*, 2150–2160. [CrossRef]
19. Mao, Z. Throughput optimization based joint access point association and transmission time allocation in WLANs. *IEEE Open J. Commun. Soc.* **2021**, *2*, 899–914. [CrossRef]
20. Kim, J.; Kim, S.H.; Sung, D.K. Hybrid ARQ-based fairness enhancement in uplink WLAN. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 4362–4373. [CrossRef]
21. Chen, Q.; Liang, G.; Weng, Z. A target wake time-based power conservation scheme for maximizing throughput in IEEE 802.11 ax WLANs. In Proceedings of the IEEE 25th International Conference on Parallel and Distributed Systems: ICPADS 2019, Tianjin, China, 4–6 December 2019; pp. 217–224.
22. Khorov, E.; Krasilov, A.; Liubogoshchev, M.; Tang, S. SEBRA: SAND-enabled bitrate and resource allocation algorithm for network-assisted video streaming. In Proceedings of the 13th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications WiMob, Rome, Italy, 9–11 October 2017; pp. 1–8.
23. Yagi, T.; Murase, T. Frame aggregation control for high throughput and fairness in densely deployed WLANs. In Proceedings of the 13th International Conference on Ubiquitous Information Management and Communication (IMCOM), Phuket, Thailand, 4–6 January 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 42–53.
24. The ns-3 Network Simulator. Available online: http://www.nsnam.org/ (accessed on 10 June 2024).
25. Obata, H.; Takano, C. Switching Method of Media Access Control for Improving Fairness of Throughput between WLAN systems under Adjacent Channel Interference. In Proceedings of the 2021 Ninth International Symposium on Computing and Networking Workshops (CANDARW), Matsue, Japan, 23–26 November 2021; pp. 42–48. [CrossRef]
26. Kong, Z.; Wu, D.; Jin, X.; Cen, S.; Dong, F. Improved AP Deployment Optimization Scheme Based on Multi-objective Particle Swarm Optimization Algorithm. *KSII Trans. Internet Inf. Syst.* **2021**, *15*, 1568–1589. [CrossRef]
27. Ghafoor, U.; Ashraf, T. Maximizing throughput and energy efficiency in 6G based on phone user clustering enabled UAV assisted downlink hybrid multiple access HetNet. *Telecommun. Syst.* **2024**, *85*, 563–590. [CrossRef]
28. Blobel, J.; Menne, F.; Yu, D.; Cheng, X.; Dressler, F. Low-power and Low-delay WLAN using Wake-up Receivers. *IEEE Trans. Mob. Comput.* **2020**, *21*, 1739–1750. [CrossRef]
29. Ali, R.; Sohail, M.; Almagrabi, A.; Musaddiq, A.; Kim, B.B. Green MAC Protocol: A Q-Learning-Based Mechanism to Enhance Channel Reliability for WLAN Energy Savings. *Electronics* **2020**, *9*, 1720. [CrossRef]
30. Dong, P.; Shen, R.; Li, Y.; Nie, C.; Xie, J.; Gao, K.; Zhang, L. An Energy-Saving Scheduling Algorithm for Multipath TCP in Wireless Networks. *Electronics* **2022**, *11*, 490. [CrossRef]
31. Karmakar, R.; Chattopadhyay, S.; Chakraborty, S. Novel AP association and fair channel access in high throughput WLAN for energy efficiency. *Ad Hoc Netw.* **2020**, *103*, 102156. [CrossRef]
32. Qiu, S.; Chu, X.; Leung, Y.W.; Ng, J.K.Y. Joint access point placement and power-channel-resource-unit assignment for 802.11 ax-based dense WiFi with QoS requirements. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 2569–2578.
33. Faria, D.B. Modeling Signal Attenuation in IEEE 802.11 Wireless LANs—Vol. 1. 2005. Available online: http://www-cs-students.stanford.edu/~dbfaria/files/faria-TR-KP06-0118.pdf (accessed on 10 June 2024).
34. Lu, T.; Funabiki, N.; Munene, K.I.; Sudibyo, R.W. An improved throughput estimation model for concurrent communications of multiple hosts in wireless local-area network. In Proceedings of the IEEE Hiroshima Section Student Symposium (HISS), Okayama, Japan, 30 November–1 December 2019; pp. 360–363.
35. Funabiki, N.; Taniguchi, C.; Lwin, K.S.; Zaw, K.K.; Kao, W.C. A parameter optimization tool and its application to throughput estimation model for wireless LAN. In Proceedings of the Conference on Complex, Intelligent, and Software Intensive Systems, Turin, Italy, 10–13 July 2017; pp. 701–710.
36. Roy, S.C.; Funabik, N.; Munene, K.I.; Rahman, M.M.; Kuribayashi, M. An extension of active access-point configuration algorithm to IEEE 802.11n and 11ac dual interfaces in wireless local-area network. *Int. J. Future Comput. Commun.* **2022**, *11*, 18–26. [CrossRef]
37. Wolsey, L.A. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica* **1982**, *2*, 385–393. [CrossRef]
38. Williamson, D.P.; Shmoys, D.B. *The Design of Approximation Algorithms*, 1st ed.; Cambridge University Press: Cambridge, UK, 2011.
39. Traffic Command, Manipulate Traffic Control Settings. Available online: https://linux.die.net/man/8/tc (accessed on 10 June 2024).

40. Hostapd. Available online: https://w1.fi/hostapd/ (accessed on 24 June 2024).
41. Archer-t4u. Available online: https://www.tp-link.com/jp/home-networking/adapter/archer-t4u/ (accessed on 24 June 2024).
42. The Ultimate Speed Test Tool for TCP, UDP and SCTP. Available online: https://iperf.fr/ (accessed on 10 June 2024).