PAPER A WDS Clustering Algorithm for Wireless Mesh Networks

Shigeto TAJIMA^{†a)}, Nobuo FUNABIKI^{††}, and Teruo HIGASHINO^{†††}, Members

SUMMARY Wireless mesh networks have been extensively studied as expandable, flexible, and inexpensive access networks to the Internet. This paper focuses on one composed of multiple access points (APs) connected through multihop wireless communications mainly by the *wireless distribution system (WDS)*. For scalability, the proper partition of APs into multiple *WDS clusters* is essential, because the number of APs in one cluster is limited due to the increasing radio interference and control packets. In this paper, we formulate this *WDS clustering problem* and prove the *NP-completeness* of its decision version through reduction from a known *NP-complete* problem. Then, we propose its heuristic algorithm, using a greedy method and a variable depth search method, to satisfy the complex constraints while optimizing the cost function. We verify the effectiveness of our algorithm through extensive simulations, where the results confirm its superiority to the existing algorithm in terms of throughput.

key words: wireless mesh network, WDS clustering, gateway selection, NP-complete, heuristic algorithm, variable depth search

1. Introduction

The Internet has dramatically evolved due to rapid development of inexpensive devices and high-speed communication technology. A variety of information, data, and services have been provided through the Internet, which has raised strong demands for Internet access from any place at any time using wireless interfaces at cellular phones, mobile personal computers (PCs) and personal digital assistants (PDAs). To meet them, wireless local area networks (WLANs) have been widely deployed as Internet-access networks. However, the area covered by WLAN with a single access point (AP) is limited to a small space because of its short transmission range. Hence, wireless mesh networks have emerged as an attractive technology [1]-[3] to expand the coverage area flexibly and inexpensively by adopting wirelessly-connected multiple APs, and they have a great potential for large-scale Internet-access networks with low costs and short deploying time.

Among several variations of wireless mesh networks being studied, we have focused on the one using APs connected mainly by the *wireless distribution system (WDS)*.

[†]The author is with the Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560–8531 Japan.

^{††}The author is with the Graduate School of Natural Science and Technology, Okayama University, Okayama-shi, 700–8530 Japan.

^{†††}The author is with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565–0871 Japan.

a) E-mail: tajima@ics.es.osaka-u.ac.jp

DOI: 10.1587/transinf.E93.D.800



Fig. 1 An WIMNET topology.

At least one AP in the network acts as a gateway to the Internet, where all traffic to/from the Internet must pass through it. For convenience, this network is called the *wireless Internet-access mesh network (WIMNET)* in this paper. Figure 1 illustrates a WIMNET topology. To reduce interference among wireless links, the IEEE802.11a protocol at 5 GHz is adopted to links between APs, and the 802.11 b/g protocol at 2.4 GHz is to links between hosts and APs. Each protocol has several non-interfered frequency channels.

For the Internet access, the packets from hosts, if associated with APs other than gateways, must reach a gateway through multihop communications between APs. As the network size increases, the links between APs around gateways become very crowded, and this might create a bottleneck for the entire communication system. To avoid this problem, we have studied several combinatorial optimization problems and their algorithms for the optimal design of WIMNET. In [4], [5], we defined an AP allocation problem and presented its heuristic algorithm. In [6], we studied a gateway AP selection problem with its algorithm. In [7], we studied a channel configuration problem with its algorithm.

In WIMNET, a set of APs behave like a single AP using WDS, called the *WDS cluster* in this paper. Due to the increase in radio interference on the same channel and broadcasting control packets among links between APs, the number of APs in a WDS cluster should be limited. Actually, this *WDS cluster size* in commercial APs is limited to six [8] or eight [9]. Hence, the proper clustering of APs into WDS clusters is essential to deploy a large-scale WIMNET composed of many APs to cover the wide area. Furthermore, this WDS clustering is effective to disperse traffic concentrations around gateways.

The proper WDS clustering is basically a hard task because it must consider several constraints and optimization

Manuscript received August 3, 2009.

Manuscript revised October 30, 2009.

indices simultaneously. The first constraint is that the number of APs in a WDS cluster must not exceed the upper limit. The second one is that all APs in a cluster must be connected with each other. The third one is that one AP in a cluster must be selected as the gateway that can deploy wired connections to the Internet. The fourth one is that the number of hosts associated with APs in a cluster must not exceed the limit, so that any cluster can ensure the communication bandwidth of hosts. As the optimizing indices, the number of WDS clusters should be minimized to save installation and operation costs of the network. The communication delay between any AP and a gateway in any cluster should be minimized to enhance the performance. Thus, the APs, the gateway, and the communication routes between APs and the gateway in every WDS cluster must be found simultaneously. As a result, the algorithm study for the WDS clustering problem becomes another critical mission for the optimal WIMNET design.

In this paper, we first formulate the WDS clustering problem for WIMNET, and prove the NP-completeness of its decision version through reduction from the NPcomplete bin packing problem [10]. Then, we propose a heuristic algorithm using a greedy method and a variable depth search (VDS) method [12] to satisfy the complex constraints while optimizing the cost function in the problem. We verify the effectiveness of our proposal through simulations using the WIMNET simulator [4], [5] developed by our group. The comparisons with the most analogous algorithm to our problem support the superiority of our approach.

The rest of this paper is organized as follows: Section 2 introduces some related works to this study. Section 3 formulates the WDS clustering problem. Section 4 gives the proof of NP-completeness. Section 5 presents our algorithm. Section 6 shows simulation results for evaluations. Section 7 provides concluding remarks with future works.

2. Related Works

In this section, we introduce several related works to the WDS clustering problem. Unfortunately, none of them deal with the four constraints in this problem including the WDS cluster size constraint at the same time. Thus, they cannot be used for the design of a large-scale WIMNET composed of multiple WDS clusters.

In [3], Aoun et al. proposed a recursive dominating set algorithm based on [11] to find a clustering such that the maximum hop count, or radius, inside a cluster is smaller than the given limit. It first extracts a dominating set of the network, and generates a graph composed of this set and the edges connecting the two APs with two hops in the network. Then, it again extracts its dominating set, where any AP is connected with three hops to an AP in this set. This recursive procedure is repeated until the hop count surpasses the limit. This algorithm cannot generate clusters with an arbitrary hop count, and cannot always satisfy the constraints of the cluster size, the bandwidth, and the gateway. In [13], Lakshmanan et al. presented a multiple gateway association model of allowing each host to be connected through more than one gateways to the Internet. They discuss its benefits in capacity, fairness, reliability, and security with its challenges. They presented the architecture using a super gateway that controls the whole system, which can be a bottleneck, and the algorithms for the gateway association and the packet transmission scheduling, which are just theoretical.

In [14], Li et al. proposed a grid-based gateway deployment method with a linear programming for a feasible interference-free TDMA link scheduling to maximize the throughput. By evaluating the throughput using the scheduling algorithm for every possible combination of Kgrid points in the field, the best locations of K gateways are found. Their method can be extended to multi-channel and multi-radio networks. However, it assumes impractical TDMA operations for wireless mesh networks. Furthermore, it does not consider the constraints of the bandwidth, the cluster size, and the connection.

In [15], Park et al. proposed a mesh router discovery scheme, and a QoS-driven mesh router selection mechanism for the dynamic gateway selection by the traffic load. In [16], Nandiraju et al. proposed a dynamic gateway selection method for load balancing among multiple gateways. Unfortunately, it does not consider interference. These methods do not intend the allocation of gateways.

In [17], Hsiao et al. proposed a multiple network composition method with the same channel by using directional antennas. In their method, a lot of APs are necessary in the field so that each host can select its associated AP from multiple candidates for load balancing.

In [18], Huang et al. investigated AP deployments for intelligent transportation systems (ITS). They proposed an optimization algorithm of a mixed-integer nonlinear programming to determine the optimal number of APs in a cluster and the best cell radius for each AP. Because their proposal targets ITS, each cluster is composed of arrayed APs and the first AP becomes the gateway.

In [19], Alicherry et al. formulated the joint problem of the channel assignment, the routing, and the scheduling for a special case of the wireless mesh network where every link activation was synchronously controlled by a single global clock, and presented its approximation algorithm that guarantees the order of approximation. Unfortunately, the realization of the synchronous wireless mesh network is very hard, and the superiority is actually not clear to the conventional asynchronous one including this paper. Furthermore, it assumes that every AP has the same number of associated hosts.

In [20], Denko studied the wireless mesh network with mobile Internet gateways using a multi-path routing scheme to increase the reliability and performance. However, the mobile gateway is not practical because the wired connection to the Internet is static. Furthermore, the network may not work properly if the traffic of every router increases, because each router selects one route by the amount of its traffic. In [21], Tokito et al. proposed a routing method for multiple gateways in wireless mesh networks, called the gateway load balanced routing (GLBR). GLBR reduces loads of congested gateways by changing the gateway of a leaf node in the routing tree one by one, such that the new gateway decreases the variance of loads at gateways and the length of the detouring path is shorter than the threshold. The initial routing tree is found by the shortest path algorithm. They show the advantage of their proposal over the shortest path routing in simulations. However, because this algorithm can change the path for only one leaf node at one time, it can be easily trapped into a local minimum where simultaneous changes of multiple paths are often necessary to escape from.

In [22], Ito et al. studied a method of distributing traffics among multiple gateways on a session by session basis in wireless mesh networks. Their method first estimates the throughput for each gateway from the traffic volume around there and the hop count, and then, selects the gateway expecting the highest throughput. Through simulations using the network simulator *ns*-2, they show the effectiveness of their proposal by comparing the throughput and the fairness between the proposed session-distribution method and the packet-distribution method.

3. Formulation of WDS Clustering Problem

3.1 Outline of WDS Clustering Problem

In the WDS clustering problem, we assume that the locations of the APs with battery supplies and the wireless links between APs in the network field have been given manually, or by using their corresponding algorithms during the design phase of WIMNET, as inputs. Thus, the topology of the AP network is described by a graph G = (V, E), where a vertex in V represents an AP and an edge in E represents a link. Each vertex is assigned the maximum number of hosts associated with the AP as the load, and each edge is assigned the transmission speed for the delay estimation, which are given as design parameters. A subset of V are designated as gateway candidates, where wired connections are available for the Internet access. The number of WDS clusters K greatly affects the installation and operation costs of WIMNET because the costly Internet gateway is necessary in each cluster. Thus, the number of WDS clusters K is also given in the input, so that the network designer can decide it. Furthermore, the limit on the cluster size and the required bandwidth in one cluster are given to determine their constraints. As the outputs of the problem, the APs, a gateway, and the communication routes between the APs and the gateway in every WDS cluster must be found simultaneously by satisfying the four constraints while optimizing the indices in Sect. 1.

3.2 Formulation of WDS Clustering Problem

Now, we formulate the WDS clustering problem for WIM-NET as a combinatorial optimization problem.

- **Input:** G = (V, E): a network topology with *N* APs $(N = |V|), h_i$: the maximum number of hosts associated with the AP_i (the *i*-th AP) for $i = 1, \dots, N$, s_{ij} : the transmission speed of the *ij*-th link $(link_{ij})$ from AP_i to AP_j in $E, X (\subseteq V)$: a set of gateway candidates, *K*: the number of WDS clusters, *H*: the limit on the number of associated hosts in a WDS cluster (bandwidth limit), and *P*: the limit on the number of APs in a WDS cluster (cluster size limit).
- **Output:** $C = \{C_1, C_2, \dots, C_K\}$: a set of WDS clusters, g_k : the gateway in C_k for $k = 1, \dots, K$, and r_i : the communication route between AP_i and the gateway.
- **Constraint:** to satisfy the following four constraints:
 - the number of APs in any WDS cluster must be *P* or smaller: $|C_i| \le P$ (cluster size constraint),
 - the number of associated hosts in any WDS cluster must be *H* or smaller: $\sum_{j \in C_i} h_j \le H$ (bandwidth constraint),
 - the APs must be connected with each other in any WDS cluster (connection constraint),
 - one gateway must be selected from gateway candidates in X in any WDS cluster (gateway constraint).
- **Objective:** to minimize the following cost function *F_c*:

$$F_{c} = A \cdot \max(hop(AP_{i})) + B \cdot \max(host(link_{ij})) + \sum_{k|eintf(ij)} host(link_{kl}))$$
(1)

where (a) A and B are constant coefficients, (b) the function max(x) returns the maximum value of x, (c) the function $hop(AP_i)$ returns the number of hops, or hop count, between AP_i and its gateway, (d) the function $host(link_{ij})$ returns the number of hosts using $link_{ij}$ in the shortest route to the gateway to represent the link load, and (e) the function intf(ij) returns the link indices that may occur the primary conflict with $link_{ij}$. The A-term represents the maximum hop count, and the B-term does the maximum total load of a link and its primarily conflicting links. The minimization of the A-and B-terms intends the maximization of the network performance.

4. Proof of NP-Completeness for WDS Clustering

In this section, the *NP-completeness* of the decision version of the WDS clustering problem (*WDS clustering*) is proved

through reduction from the *NP-complete* bin packing problem (*Bin packing*) [10].

4.1 Decision Version of WDS Clustering Problem

WDS clustering is defined as follows:

- **Instance:** The same inputs as the WDS clustering problem with an additional constant F_{c0} .
- Question: Is there a WDS clustering with *K* clusters to satisfy *F_c* ≤ *F_{c0}* ?

4.2 Bin Packing

Bin packing is defined as follows:

- **Instance:** $U = \{u_1, u_2, \dots, u_{|U|}\}$: a set of items with various volumes, and *L* bins with a constant volume *B*.
- **Question:** Is there a way of partitioning all the items into the *L* bins ?

4.3 Proof of NP-Completeness

Clearly, *WDS clustering* belongs to the class *NP*. Then, an arbitrary instance of *Bin packing* can be transformed into the following instance of *WDS clustering*. Thus, the NP-completeness of *WDS clustering* is proved.

- Input: $G = (V, E) = K_N$: a complete graph with N = |V| = |U|, $s_{ij} = 1$, $h_i = u_i$ for $i = 1, \dots, N$, X = V, H = B, $P = \infty$, K = L, and $F_{c0} = \infty$.
- **Output:** The set of WDS clusters is equivalent to the bin packing, where any AP can be a gateway and is one-hop away from the gateway in each WDS cluster.
- **Constraint:** to satisfy the following four constraints:
 - the number of APs in any WDS cluster is not limited $(P = \infty)$,
 - the number of associated hosts in any WDS cluster must be H = B or smaller,
 - the APs are connected with each other in any WDS cluster ($G = K_N$), and
 - the gateway is selected from gateway candidates in any WDS cluster (X = V).
- Objective: The condition F_c ≤ F_{c0} is always satisfied with F_{c0} = ∞.

5. WDS Clustering Algorithm

In this section, we propose a heuristic algorithm for the WDS clustering problem to avoid combinatorial explosions. As an efficient heuristic, our algorithm finds an initial solution by a greedy method, and then, improves it by a VDS method that can enhance the search ability of a local search method by expanding neighbor states flexibly.

Our algorithm seeks the maximization of the network performance with the number of clusters K. If any feasible solution cannot be found with this number, our algorithm terminates after reporting the failure.

5.1 Check of Number of Clusters

First, the feasibility of the number of clusters K in the input is checked, because it has the trivial upper and lower limits that can be given by other inputs of the problem. The upper limit K_{max} is given by the number of gateway candidates: $K_{\text{max}} = |X|$. The lower limit K_{min} is given by the following equation to satisfy the cluster size constraint and the bandwidth constraint:

$$K_{\min} = \max\left\{ \lceil N/P \rceil, \left[\sum_{i=1}^{N} h_i / H \right] \right\}$$
(2)

where the ceiling function $\lceil x \rceil$ returns the smallest integer *x* or more. Then, if $K < K_{\min}$ or $K > K_{\max}$, our algorithm terminates after reporting the feasible range of *K*.

5.2 Initial Gateway Selection

In our algorithm, *K* APs are randomly selected as initial gateways among gateway candidates in *X* such that two selected APs are not adjacent to each other as best as possible. Starting from these selected APs, the initial WDS clusters are constructed sequentially. Then, the WDS clusters are iteratively improved by the *Variable Depth Search* (*VDS*) method. This WDS clustering procedure is repeated by min(2N, $|_{X|}C_K$) times because initial gateway APs are selected by different combinations, and the best solution in terms of the cost function is selected as the final solution.

5.3 Greedy Construction

Our algorithm generates an initial WDS clustering by repeating the following procedure:

- 1. Sort the APs adjacent to the clustered APs in descending order of its load h_i . If two or more such APs have the same load, resolve this tiebreak in ascending order of the number of incident links.
- 2. Apply the following procedure for each AP in step 1 from the top:
 - a. Select the cluster of its adjacent AP as a cluster candidate for the AP, if the following two constraints are satisfied:
 - the number of APs in the cluster is smaller than *P* for the cluster size constraint, and
 - the total number of associated hosts in the cluster is *H* or smaller after the clustering for the bandwidth constraint.
 - b. Cluster the AP as follows, if at least one cluster candidate is selected.
 - Select this cluster candidate if only one candidate exists, or otherwise
 - Select the cluster candidate that minimizes the cost function F_c .

3. Repeat steps 1–2 until every AP is clustered or no more AP can be clustered.

5.4 Gateway Update

If the selected AP in the sequential AP clustering (let AP_k) is a gateway candidate, the shortest path is calculated from every AP in the same cluster to this AP passing through only APs in this cluster, and the following gateway cost function F_k is computed:

$$F_k = \max(host(link_{ij})).$$
(3)

If F_k becomes smaller, AP_k is selected as the new gateway in the corresponding cluster.

5.5 Local Search by VDS

Then, the initial WDS clustering is improved iteratively by repeating the cluster changes of multiple APs at the same time using the *variable depth search (VDS)* method. VDS is a generalization of a local search method, where the size of neighborhood is adaptively changed so that the algorithm can effectively traverse the large search space while keeping the amount of computational time reasonable. Actually, because each feasible state in this problem may have a different size of its neighborhood that satisfies the four constraints, VDS is suitable for this problem.

In our VDS for the WDS clustering, a simple *move* operation is repeatedly tried until no further feasible operation is possible. Each *move* operation changes the cluster of an AP into a different feasible one such that the cost function F_c in (1) becomes minimum among the candidates. Then, only the subsequence of the *move* operations resulting into the smallest cost function are selected to be actually applied there, only if F_c after these operations becomes equal to or smaller than that of the previous state. If the cluster of any AP is not changed at one iteration or the cost function has not been improved during *R* iterations (R = 10 in the paper), the state is regarded as the local minimum. Then, the hill-climbing procedure is applied for the state to escape from it.

When the hill-climbing procedure is applied in T times, the local search by VDS is terminated (T = 20 in the paper), and the best found solution is output as the final one. At this time, if an AP is not clustered at all, our algorithm regards that the K clustering of the APs is impossible and terminates after reporting the failure.

In summary, one iteration of this stage consists of three steps: 1) the cluster change trial, 2) the cluster change application, and 3) the hill-climbing. The following three sections describe them in detail. Here, we note that the unclustered APs in the initial WDS clustering may be clustered in VDS.

5.5.1 Cluster Change Trial

The cluster change trial repeats the cluster change of the AP

that satisfies the following three conditions until no more change is possible:

- 1. the AP has not been selected at this iteration,
- 2. the resulting clustering satisfies the constraints, and
- 3. the resulting clustering minimizes the cost function F_c among candidates.

5.5.2 Cluster Change Application

The cluster change trial always changes the AP cluster regardless of the increase of the cost function F_c as long as it satisfies the constraints. Thus, F_c may increase after some cluster changes. The *cluster change application* selects the subsequence of the cluster changes that minimizes F_c , and actually apply these cluster changes with the gateway update procedure in Sect. 5.4 to the current solution, only if F_c becomes equal or smaller than that of the previous iteration. If the cluster changes are actually applied, another iteration is repeated from the cluster change trial.

5.5.3 Hill Climbing

The local search process using *move* operations in our VDS may be trapped into a local minimum where the solution cannot be improved without the hill-climbing step. In our algorithm, when either of the following two conditions is satisfied in Sect. 5.5.2, the current state is regarded as a local minimum, and the hill-climbing procedure is applied to escape from it:

- 1. no cluster change is applied at one iteration, or
- 2. F_c has not been improved during *R* iterations (R = 10 in the paper).

In the hill-climbing procedure, the following *random* cluster change operation is repeated until the clusters of S APs are actually changed, or no more APs can be changed (S = 10 in the paper).

- 1. Enumerate any AP that satisfies the following three conditions for the random cluster change:
 - a. it is not selected at this hill-climbing procedure,
 - b. it is located on the boundary between different clusters, and
 - c. its cluster change does not affect the connectivity of the other APs in the same cluster.
- 2. Randomly select one AP among them.
- 3. For this AP, find any cluster that can feasibly be changed into.
- 4. If such a cluster exists, change the cluster of this AP to a randomly selected cluster among them.
- 5. Otherwise, remove the cluster of this AP.

6. Performance Evaluation by Simulations

In this section, we discuss the performance evaluation of

our algorithm for the WDS clustering problem through network simulations using the WIMNET simulator. For this evaluation, the compared algorithm in Sect. 6.2 is also implemented. In each simulated instance, the minimum number of clusters such that each algorithm can find a feasible solution is given for the number of clusters K respectively, because we regard the minimization of K as the first priority task in the WIMNET design to reduce the installation and operation costs.

6.1 WIMNET Simulator

The WIMNET simulator implements the least functions for simulating wireless communications to evaluate throughputs and delays of a large-scale WIMNET with reasonable CPU time on a conventional PC. Thus, a sequence of functions such as host movements, communication request arrivals, and link activations are synchronized by a single global clock called a time slot. Within an integral multiple of time slots, each host/AP can complete the one-packet transmission and the acknowledgment reception of a link. In our simulations, the duration time of one time slot is set 0.2 ms. From our experimental results, the maximum transmission speed between APs is set 30 Mbps, and that between an AP and its associated host is 20 Mbps. Thus, the former link is completed with two slots, and the latter is with three slots, assuming every packet has 1,500 bytes. When two or more links within their interference ranges may be activated at the same time slot, randomly selected one link among them is successfully activated, and the others are inserted into waiting queues to avoid collisions, supposing DCF and **RTS/CTS** functions.

Before starting each simulation, every host has 125 packets transmitted to the gateway, and the gateway has 1,000 packets to each host. When every packet reaches the destination or is lost, the simulation is finished. Note that actually no packet loss occurs in our simulations, assuming every AP has a finite size of queues. The packets of each request are transmitted along the communication route found by our algorithm. Only the connection-less communication is implemented, where retransmissions between end hosts are not considered.

6.2 Compared Algorithm

Within our knowledge, no algorithm has been reported for the same WDS clustering problem in this paper. Therefore, as the most analogous algorithm to our problem, the *Open/Close method* in [2] has been implemented with some modifications for performance comparisons with our algorithm, where it does not consider the cluster size constraint and the distribution of associated hosts with APs. The procedure of this heuristic algorithm is described as follows.

6.2.1 Initial WDS Clustering

1. Generate the sorted list of the APs in descending order

of the maximum number of associated hosts.

- 2. Select the first *K* APs in the list as gateways.
- 3. Assign the WDS cluster to an unclustered AP that satisfies the following conditions:
 - the AP is adjacent to an AP clustered to this WDS cluster,
 - the cluster size constraint is satisfied if added,
 - the bandwidth constraint is satisfied if added, and
 - the hop count (the number of hops between the AP and the gateway) is minimized.
- 4. Repeat step 3 until no more AP can be assigned.
- 5. Calculate the sum of the hop count of every AP, if every AP is assigned a cluster, and save it.

6.2.2 WDS Clustering Improvement

The initial clustering is iteratively improved by repeating the following three operations:

- 1. Close operation
 - a. Remove one gateway randomly, and uncluster all the APs connected to this gateway.
 - b. Go to *Open operation*.

2. Open operation

- a. Select the first AP of the list in Sect. 6.2.1 as the gateway that has not been selected.
- b. If no more AP is selected in step a, output the bestfound solution if found, or output the error otherwise, and terminate the procedure.
- c. Assign the WDS cluster to an unclustered AP that satisfies the four conditions in Sect. 6.2.1.
- d. Repeat step c until no more AP can be assigned.
- e. If every AP is assigned a cluster, calculate the sum of the hop count of every AP, and save it if the value is smaller than the best-found one. Return to *Close operation*.
- f. Otherwise, go to Cluster adjustment.

3. Cluster adjustment

- a. Assign the unassigned AP to one of the connectable WDS clusters randomly.
- b. If the cluster size constraint or the bandwidth constraint is not satisfied as the result of the assignment in step a, APs in the cluster are unclustered one by one in ascending order of the hop count until the constraint is satisfied. If every AP in the cluster except the gateway is unclustered but the constraint is not still satisfied, every unclustered AP is resumed and the cluster assignment in step a is discarded.
- c. If every AP is assigned a cluster, calculate the sum of the hop count of every AP, and save it if the value is smaller than the best-found one. Return to *Close operation*.

d. If no feasible solution is obtained after repeating *Cluster adjustment* in 300 times, abort the procedure, and return to *Close operation*.

We note that the original Open/Close method assumes that each gateway may have a different bandwidth for communications to/from wired networks to the Internet. In our implementation, we use the maximum number of associated hosts with an AP as this bandwidth.

6.3 Simulations for Different Traffic Patterns

In our first simulations, the performance of our algorithm is evaluated through simple instances whose optimal solutions can be found easily, so that the optimality of our heuristic algorithm can be verified. For this purpose, we adopt the simple network topology of regularly allocated $24 (= 6 \times 4)$ APs, where each AP has wireless links with its four neighbor APs on the left, right, top, and bottom sides. This grid topology has been often used in wireless mesh network studies [19], [23]–[26]. To generate non-uniform traffics using simple loads, 8 APs among 24 are associated with 10 hosts, and the remaining 16 APs are with 1 host, which means the total of 96 hosts exist in the network. Then, by changing the locations of crowded APs in the field, we prepare 10 instances of different traffic patterns.

As the input parameters of the algorithm, the cluster size limit P is set 6 and the bandwidth size limit H is 24 where the lower limit on the number of clusters K_{\min} is 4. Every link is assigned the same bandwidth $s_{ii} = 30$ Mbps, and every AP becomes a gateway candidate with X = Vfor simplicity. The coefficients A = B = 1 are used for the cost function F_c , because our preliminary experiments using these instances observed no big difference in throughputs when A and B were changed from 1 to 3. To avoid the bias in random numbers, the average result among 10 runs using different random numbers is used in the evaluation for each instance. As example instances in our first simulations, Fig. 2 illustrates traffic patterns and our clustering results with four clusters (K = 4) for four instances among them, where a black circle represents an AP associated with 10 hosts, and a white one represents an AP with 1 host. These results are actually optimum in these instances with the minimum number of clusters and cost functions.

Figure 3 compares the average number of clusters between two algorithms for each of 10 instances. Our algorithm ("Proposal") always finds a feasible solution with the minimum number of clusters for any instance, whereas the compared one ("Comparison") usually requires larger numbers. The reason may come from the fact that our algorithm seeks a feasible better solution with the fixed number of clusters, whereas the compared one does not explicitly minimize the number of clusters and may reduce it by chance through repeating open/close operations.

Then, to evaluate the WDS clustering results in terms of the network performance, the WIMNET simulator is applied using the clustering results by both algorithms. Fig-



Fig. 2 Four traffic patterns and clustering results in first simulations.



Fig. 3 Average number of clusters for different traffic patterns.



Fig. 4 Average throughputs for different traffic patterns.

ure 4 compares the average total throughput for each instance between two algorithms, where our algorithm provides the larger throughput than the compared one by 24%– 80% for any instance. Here, we analyze the reason why our algorithm achieves at least 150 Mbps. The total throughput of one WDS cluster is determined by the summation of the gateway throughput and the maximum communication throughput between APs in WIMNET. As shown in Fig. 3, the traffic load is evenly distributed among four clusters in our algorithm, which gives the same throughput for every cluster. As a result, the total throughput of 150 Mbps or more comes from the formula of $((30 + \Delta) \times 4)$ Mbps where Δ represents the gateway throughput.



Fig. 5 Performance comparison of *F_c* with and without A or B-term.

6.4 Simulations for Verification of Terms in Cost Function

The importance of each term in the cost function F_c is verified through simulations using the 10 instances in Sect. 6.3. Figure 5 compares the average throughput among the four different conditions for F_c , where *AB* represents the result using both terms, *A* does the result using the *A*-term only, *B* does the result using the *B*-term only, and *None* does the result without using F_c . This figure indicates that *AB* provides the best throughput in any simulated instance. Note that all of them find the solution with the least number of clusters. Thus, we conclude that the two terms in the cost function F_c are necessary for finding the high quality WDS clustering.

6.5 Simulations for Different Bandwidth Limits

In our second simulations, the performance for different bandwidth limits is investigated for instance 1 in Fig. 2. *P* is fixed with 8, and *H* is selected between 21 and 48, where K_{\min} is 3, 4, or 5. Figures 6 and 7 compare the average number of clusters and the average total throughput, respectively. The number of clusters by our algorithm is always smaller than that by the compared one, and the throughput is larger by 10%–183%. Generally, as the bandwidth constraint becomes harder, both the number of clusters and the average throughput increase except for H = 21.

6.6 Simulations for Different Number of Clusters

In our third simulations, the performance for different number of clusters is investigated using instance 4 in Fig. 2, where P = 12 and H = 48 are used, and the number of clusters K is changed from 2 to 24. Figure 8 shows changes of the throughputs by two algorithms and the cost function F_c in our algorithm. This result indicates that as K increases until certain values, F_c decreases and the throughput increases in our algorithm, and the throughput by our algorithm is always better than that by the compared algorithm when it is not saturated. The results confirm the effectiveness of our algorithm for different number of clusters.

Here, we note that F_c and the throughput are saturated at certain values of K. The reason is that the bottleneck



Fig. 6 Average number of clusters for different bandwidth limits.



Fig. 7 Average throughputs for different bandwidth limits.



Fig.8 Relationship between Network performance and minimum value of F_c by K.

WDS cluster cannot be changed there due to the constraints of the problem, or the communication bandwidth between an AP and a host (20 Mbps in simulations) becomes the bottleneck of the whole communications.

6.7 Simulations for Sparse Networks

In our fourth simulations, the performance for sparse networks is investigated with the same number of APs. Here, the same 24 APs and traffic patterns in the first simulations are adopted, where the links are randomly generated with the 50% probability with additional links for the connectivity. As the input parameters of the algorithm, P = 6 and



Fig. 9 Average number of clusters for sparse networks.



Fig. 10 Average throughputs for sparse networks.

H = 35 are used for $K_{\min} = 4$. Figures 9 and 10 compare the average number of clusters and the average total throughput for each instance. The number of clusters found by our algorithm is smaller than that by the compared one. The throughput by our algorithm is larger by 23%-143% than the compared one except instance 10. In instance 10, the average throughput by the compared algorithm is larger than that by our algorithm, because its number of clusters (K = 5) is large than ours (K = 4). We note that if the number of clusters is the same, the throughput by the compared algorithm is compared algorithm is smaller than that by ours.

6.8 Simulations for Random Networks

In our last simulations, the performance for random networks with 50 APs is investigated to evaluate our algorithm in more practical situations. The APs are randomly allocated on the network field $(500 \text{ m} \times 500 \text{ m})$ in the paper) such that the distance between any pair of APs is larger than the minimum one (50 m) in the paper). Then, the wireless link is generated for any pair of APs within the distance of 110 m representing the wireless range in a free space. However, this wireless link can be blocked by obstacles such as walls and furniture in indoor environments as target fields for WIMNET. In order to consider the link failure stochastically, the following Waxman method is adopted to generate the link randomly, which has been often used in network studies [27]:



Fig. 11 Clustering results for two random networks.

$$P(u,v) = \alpha e^{-d/(\beta D)} \tag{4}$$

where P(u, v) is the probability of generating a link between AP_u and AP_v , α and β are constants satisfying $0 < \alpha, \beta \le 1$ ($\alpha = 0.9, \beta = 0.8$ in the paper), d is the distance between AP_u and AP_v , D is the largest distance between two APs in the network (on average, D = 647.6 m in the paper). Then, the maximum number of hosts associated with each AP is randomly generated between 1 and 10 such that the total number of them becomes 200, in order to consider various network situations under the constant total load. As the constraints for WDS clusters, P = 6 and H = 25 are used for $K_{\min} = 9$.

By changing random numbers, 10 topologies are generated, and WDS clusterings are found by applying both algorithms to each topology in 10 times. Then, the WIMNET simulator is executed with each WDS clustering in three times using different random numbers. As a result, the average number of clusters and throughputs among the total of 30 trials for each of 10 topologies are evaluated in random network instances. Figure 11 illustrates two topologies with WDS clusterings and gateways found by our algorithm.

Figure 12 and 13 compare the simulation results by both algorithms. The results show that our algorithm can find the WDS clustering with the least number of clusters, which provides the better performance than the compared one for practical instances.

6.9 Simulations for Load Changes in Random Networks

In the WDS clustering problem for WIMNET, the maximum number of associated hosts with each AP is given as the input. Normally, the number of associated hosts with an AP is frequently changing between 0 and this maximum number, because client hosts are often moving and are randomly connecting to the Internet through WIMNET.

In order to evaluate the performance of our algorithm in such normal situations, one random network instance is simulated when the number of associated hosts with each AP is changed randomly between the minimum and the given maximum. To vary the load, this minimum is changed from 1% of the maximum until reaching the maximum with the 1% interval. Figure 14 compares the throughputs between our algorithm and the compared one under 100 different loads. The result shows that the WDS clustering by our algorithm provides the better throughput at any load than the



Fig. 12 Average number of clusters for random networks.



Fig. 13 Average throughputs for random networks.



Fig. 14 Throughput changes for different numbers of hosts.

compared one. Here, we note that if the maximum load for an AP is changed, the WDS clustering should be redesigned by applying our algorithm.

7. Conclusion

In this paper, we defined the formulation of the WDS clustering problem in the wireless Internet-access mesh network (WIMNET) with the proof of the NP-completeness of its decision version. Then, we proposed its heuristic algorithm using a greedy method and a variable depth search method. Its effectiveness is verified through extensive network simulations using the WIMNET simulator, where the comparisons of the number of clusters and throughputs with an existing algorithm confirm the superiority of our algorithm. Our future works may include simulations with more realistic situations, the development of the distributed version of the WDS clustering algorithm, and experiments using real networks.

References

- I.F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," Comput. Network. ISDN Syst., vol.47, no.4, pp.445–487, March 2005.
- [2] R. Prasad and H. Wu, "Gateway deployment optimization in cellular Wi-Fi mesh networks," J. Networks, vol.1, no.3, pp.31–39, July 2006.
- [3] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, "Gateway placement optimization in wireless mesh networks with QoS constraints," IEEE J. Sel. Areas Commun., vol.24, no.11, pp.2127–2136, Nov. 2006.
- [4] N. Funabiki, T. Nakanishi, Y. Nomura, T. Farag, S. Tajima, and T. Higashino, "An optimal access-point allocation for wireless infrastructure mesh networks," Proc. 16th Int. Conf. Comp. Theory and Applications (ICCTA), Sept. 2006.
- [5] T. Farag, N. Funabiki, and T. Nakanishi, "An access point allocation algorithm for indoor environments in wireless mesh networks," IEICE Trans. Commun., vol.E92-B, no.3, pp.784–793, March 2009.
- [6] S. Tajima, T. Higashino, N. Funabiki, and S. Yoshida, "An Internet gateway access-point selection problem for wireless infrastructure mesh networks," Proc. Int. Work. Future Mob. Ubiquitous Inform. Technol. (FMUIT), pp.133–137, May 2006.
- [7] N. Funabiki, T. Nakanishi, W. Hassan, and K. Uemura, "A channel configuration problem for access-point communications in wireless mesh networks," Proc. Int. Conf. Networks (ICON), pp.240–245, Nov. 2007.
- [8] http://buffalo.jp/
- [9] http://www.iodata.jp/
- [10] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W.H. Freeman, 1979.
- [11] V. Chvatal, "A greedy heuristic for the set-covering problem," Math. Oper. Res., vol.4, no.3, pp.233–235, 1979.
- [12] K. Katayama, A. Hamamoto, and H. Narihisa, "An effective local search for the maximum clique problem," Inf. Process. Lett., vol.95, no.5, pp.503–511, Sept. 2005.
- [13] S. Lakshmanan, K. Sundaresan, and R. Sivakumar, "On multigateway association in wireless mesh networks," Proc. IEEE Workshop. Wireless Mesh Networks (WiMesh), pp.64–73, Sept. 2006.
- [14] F. Li, Y. Wang, and X.-Y. Li, "Gateway placement for throughput optimization in wireless mesh networks," Proc. IEEE Int. Conf. Commun. (ICC), pp.4955–4960, June 2007.
- [15] B.-N. Park, W. Lee, S. Ahn, and S. Ahn, "QoS-driven wireless broadband home networking based on multihop wireless mesh networks," IEEE Trans. Consum. Electron., vol.52, no.4, pp.1220– 1228, Nov. 2006.
- [16] D. Nandiraju, L. Santhanam, N. Nandiraju, and D.P. Agrawal, "Achieving load balancing in wireless mesh networks through multiple gateways," Proc. Mobile Adhoc and Sensor Systems (MASS), pp.807–812, Oct. 2006.
- [17] P-H. Hsiao and H.T. Kung, "Layout design for multiple collocated wireless mesh networks," Proc. Vehicular Technology Conf. (VTC), vol.5, pp.3085–3089, Sept. 2004.
- [18] J.-H. Huang, L.-C Wang, and C.-J. Chang, "Wireless mesh networks for intelligent transportation systems," Proc. Systems, Man and Cybernetics (SMC), vol.1, pp.625–630, Oct. 2006.
- [19] M. Alicherry, R. Bhatia, and L.E. Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," IEEE J. Sel. Areas Commun., vol.24, no.11, pp.1960–1971, Nov. 2006.
- [20] M.K. Denko, "Using mobile internet gateways in wireless mesh net-

works," Proc. Advanced Inform. Network. Applications (AINA), vol.1, pp.1086–1092, March 2008.

- [21] H. Tokito, M. Sasabe, G. Hasegawa, and H. Nakano, "Routing method for gateway load balancing in wireless mesh networks," Proc. Int. Conf. Networks (ICN), pp.127–132, March 2009.
- [22] M. Ito, T. Shikama, and A. Watanabe, "Proposal and evaluation of multiple gateways distribution method for wireless mesh network," Proc. Int. Conf. Ubiquitous Inform. Manage. Commun. (ICUIMC), pp.18–25, Jan. 2009.
- [23] J. Robinson and E.W. Knightly, "A performance study of deployment factors in wireless mesh networks," Proc. Inform. Commun. (INFOCOM), pp.2054–2062, May 2007.
- [24] Y. Yan, H. Cai, and S-W. Seo, "Performance analysis of IEEE802.11 wireless mesh networks," Proc. IEEE Int. Conf. Commun. (ICC), pp.2547–2552, May 2008.
- [25] L. Badia, A. Etra, L. Lenzini, and M. Zorzi, "A general interferenceaware framework for joint routing and link scheduling in wireless mesh networks," IEEE J. Network, vol.22, no.1, pp.32–38, Jan./Feb. 2008.
- [26] F. Ye, Q. Chen, and Z. Niu, "End-to-end throughput-aware channel assignment in multi-radio wireless mesh networks," Proc. Global Telecommun. (GLOBECOM), pp.1375–1379, Nov. 2007.
- [27] B.M. Waxman, "Routing of multipoint connections," IEEE J. Sel. Areas Commun., vol.6, no.9, pp.1617–1622, Dec. 1988.



Teruo Higashino received the B.S., M.S., and Ph.D. degrees in information and computer sciences from Osaka University, Japan, in 1979, 1981 and 1984, respectively. He joined the faculty of Osaka University in 1984. Since 2002, he has been a professor in Graduate School of Information Science and Technology at Osaka University. In 1990 and 1994, he was a visiting researcher at University of Montreal, Canada. His current research interests include design and analysis of distributed systems, communication

protocol, and mobile computing. Dr. Higashino is a senior member of IEEE and a fellow of IPS of Japan.



Shigeto Tajima received the B.S. degree in industrial engineering from Osaka Electro-Communication University, Japan, in 1992. From 1992 to 1995, he was with the System Engineering Division, NS and I System Service Corp., Japan. In 1995, He joined the Department of Information and Computer Sciences at Osaka University, Japan, as an experimental officer, and became an research associate in 1996. His research interests include computer network and optimization algorithm. Mr. Tajima is a

member of the IPS of Japan.



Nobuo Funabiki received the B.S. and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984 and 1993, respectively. He received the M.S. degree in electrical engineering from Case Western Reserve University, USA, in 1991. From 1984 to 1994, he was with the System Engineering Division, Sumitomo Metal Industries, Ltd., Japan. In 1994, he joined the Department of Information and Computer Sciences at Osaka University, Japan, as an

assistant professor, and became an associate professor in 1995. He stayed at University of Illinois, Urbana-Champaign, in 1998, and at University of California, Santa Barbara, in 2000–2001, as a visiting researcher. In 2001, he moved to the Department of Communication Network Engineering at Okayama University as a professor. His research interests include computer network, optimization algorithm, image processing, educational technology, Web technology, and network security. Dr. Funabiki is a member of the IEEE and the IPS of Japan.