A Smart Access-Point Selection Algorithm for Scalable Wireless Mesh Networks

Nobuo Funabiki, Sritrusta Sukaridhoto, Masaharu Hata, Shigeru Tomisato, Toru Nakanishi, Kan Watanabe, and Shigeto Tajima

***Abstract*—As a flexible, cost-efficient solution for a scalable Internet-access network, we have studied the architecture and design optimization issues of the *Wireless Internet-access Mesh NETwork (WIMNET)* that is composed of wirelessly connected multiple access-points (APs). WIMNET utilizes two types of APs as wireless mesh routers to achieve the scalability with sufficient bandwidth while reducing costs. One is an expensive, programmable *smart AP (SAP)* that can use plural channels for wireless communications and has various functions for the Internet access. Another is an inexpensive, non-programmable *conventional AP (CAP)* that can use only one channel. To enhance the performance of WIMNET with a small number of costly SAPs, the allocation of SAPs in the network field is very important. In this paper, we propose a *SAP selection algorithm* of selecting a fixed number of SAPs from a given set of allocated APs. Then, we extend this algorithm to finding the *minimal SAP set* that provides the maximal throughput for the efficient WIMNET. We verify the effectiveness of our proposals through extensive simulations using the WIMNET simulator.**

***Index Terms*—Wireless mesh network, smart access-point, selection, algorithm, maximal throughput, minimal cost**

I. Introduction

THE *wireless local area network (WLAN)* has been extensively deployed around the world as an inexpensive and flexible access network to the Internet. Because the WLAN does not need a wired cable to connect a host with an access point (AP), it has advantages over a wired LAN such as low installation and management costs, easy host relocations, and flexible service areas. An AP acts as a connection hub to a wired network in the WLAN. As a result, the WLAN has been installed at many places and organizations including governments, companies, homes, and schools. Nowadays, the WLAN service has become available even in moving public spaces such as trains and airplanes.

The WLAN, however, has a drawback such that one AP can cover only the limited area within approximately $100m$ distance due to the weak transmission signal. For the WLAN service to the wide area, multiple APs should be installed. These APs are usually connected through wired cables, whereas the cabling cost may impair the cost and flexibility advantages of the WLAN. Besides, the cable may not be able to be laid down in places such as outdoors and old buildings.

Manuscript received July 11, 2011; revised July 15, 2011. This work was supported in part by JSPS KAKENHI (22500059).

N. Funabiki, S. Sukaridhoto, M. Hata, S. Tomisato, T. Nakanishi, and K. Watanabe are with the Department of Electrical and Communication Engineering, Okayama University, Okayama 700-8530, Japan, e-mail: {funabiki,hata,tomisato,nakanisi,can}@cne.okayama-u.ac.jp.

S. Tajima is with the Department of Information and Computer Sciences, Osaka University, Toyonaka 560-8531, Japan, e-mail: tajima@ics.es.osaka-u.ac.jp.

One solution to this problem is the mesh allocation of multiple APs using wireless communications between adjacent APs in the service area, in addition to conventional wireless communications between APs and hosts. Distant APs can be communicated through multihop communications, where intermediate APs act as repeaters to reply packets. This multihop WLAN is called the *wireless mesh network* [1]-[3].

Among several variations under studies for the wireless mesh network, we have focused on the one that uses only APs as wireless mesh routers and realizes communications between APs mainly on the MAC layer with the *wireless distribution system (WDS)*. At least one AP acts as a *gateway (GW)* to the Internet, where any host can connect to the Internet through this GW. We have called it *WIMNET (Wireless Internet-access Mesh NETwork)* for convenience [3]-[13].

When the size of WIMNET is expanded for the increasing number of APs and hosts, it may meet two serious problems. One is the *increase of communication delay* due to the bandwidth shortage at the wireless links around the GW, because increasing traffics between the Internet and WIMNET must pass through them, whereas their bandwidth is limited [14]. Another is the *degradation of dependability and communication quality* due to the increasing interference between wireless links. As a result, the number of APs in a single WDS must be limited to avoid the unacceptable interference.

In order to solve these problems, we have proposed a hierarchical structure for WIMNET that is composed of two types of APs and *WDS clusters*. As shown in Figure. 1, one WDS cluster consists of one expensive, programmable *smart AP (SAP)* as the cluster head, and plural inexpensive, non-programmable *conventional APs (CAPs)*. A SAP can use plural channels for wireless communications by equipped with additional network interface cards (NICs) [15][16], and has various functions for the Internet access [6]. On the other hand, a CAP can use only one channel. The number of CAPs in one WDS cluster is limited because they need to periodically exchange the routing information for WDS. The WDS clusters are connected through SAPs using plural channels because their traffics are usually larger than traffics inside the WDS cluster. Then, the proper SAP selection among allocated APs becomes very important because it determines the performance and the cost of WIMNET.

In this paper, we first define this *SAP selection problem* of selecting the given fixed number of SAPs from a given set of allocated APs that maximizes the throughput, and present its heuristic algorithm [10]. In this algorithm, the throughput maximization is sought by checking every feasible set of

(Advance online publication: 24 August 2011)

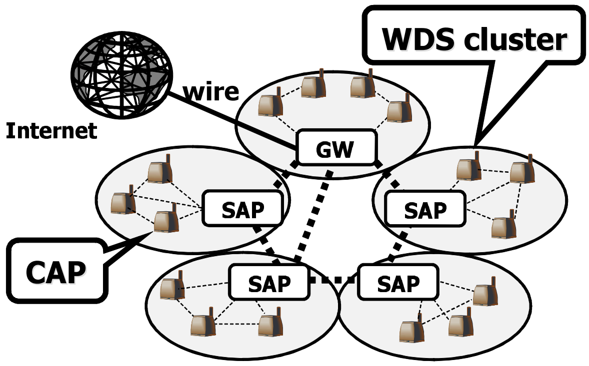

Fig. 1. Outline of WIMNET

SAPs among APs whether it minimizes the maximum delay at one channel. This delay is estimated through the summation of the traffics among the interfered links using the same channel. The effectiveness of the SAP selection algorithm is verified through simulations in four instances using the WIMNET simulator.

Then, since the simulation results for the SAP selection algorithm observed the existence of a *minimal SAP set* providing the maximal throughput, we propose the extension of the SAP selection algorithm to finding this minimal SAP set with the maximal throughput [11]. In this extension, we adopt the cost function in the SAP selection algorithm to check the maximality in terms of the throughput. The effectiveness of this algorithm extension is verified through simulations in the same four instances, where any result indicates that the SAP set by our algorithm provides the maximal throughput similar to the one where every AP becomes a SAP. We conclude that using our proposal, a large-size, high-performance WIMNET can be configured with a small number of costly SAPs.

The rest of this paper is organized as follows: Section II presents the SAP selection algorithm for WIMNET. Section III presents the extension to the minimal SAP selection for the maximal throughput. Section IV shows evaluation results by simulations. Section V concludes this paper with some future works.

II. Proposal of SAP Selection Algorithm

In this section, we define the SAP selection problem for WIMNET, and present its algorithm to seek the minimization of the cost function representing the maximum delay of one link by combining our two existing algorithms for WIMNET.

A. Definition of SAP Selection Problem

As the inputs to the SAP selection problem, we assume that the AP network topology, the GW, the expected maximum number of associated hosts with each AP as traffic loads, the interference among the links, the transmission speed of each link, the WDS cluster size limit, and the channel interference matrix are given. As the output, the SAP set with the routing tree and NIC/channel assignments is requested such that they can minimize the maximum delay as the cost function. Then, the SAP selection problem can be formulated as follows:

A. Input:

- the AP network topology: $G = (V, E)$
  - the set of APs: V
  - the number of APs: $N = |V|$
  - the set of links between APs: E
  - the interference among links: $D = [d_{ijpq}]$, where $d_{ijpq} = 1$ if two links, $AP_i \rightarrow AP_j$ and $AP_p \rightarrow AP_q$, are interfered with each other, and $d_{ijpq} = 0$ otherwise
  - the bandwidth of $link_{ij}$: s_{ij} (Mbps)
  - the expected maximum number of associated hosts with AP_i: h_i
  - the Internet gateway: g ($\in$ V)
- the number of SAPs: M
- the WDS cluster size (the maximum number of CAPs in one WDS cluster): S
- the number of channels: P
- the channel interference matrix: $C = [c(i, j)]$
- the maximum number of NICs per SAP: Q

B. Output: the SAP set with the routing tree and NIC/channel assignments

C. Constraints: The following three conditions must be satisfied in the feasible solution:

(1) The gateway must be a SAP.

(2) Any CAP must not exist along the routing path between the GW and any SAP.

(3) Any CAP must have at least one connectable SAP as the cluster head.

The *connectable SAP* represents a SAP that exists along the shortest path between the CAP and the GW, or exists within four hops from the CAP. The latter condition is necessary to increase the number of cluster head candidates for CAPs to obtain feasible solutions.

D. Objective: The cost function E representing the maximum delay should be minimized:

$$E = \max_{(i,j)} \left[t_{ij} + t_{ji} + \sum_{\substack{p=1 \\ d_{ijpq}=1 \vee d_{jipq}=1}}^{N} \sum_{q=1}^{N} t_{pq} c\left(y_{ij}, y_{pq}\right) \right] \quad (1)$$

where t_{ij} represents the traffic along the link from $AP_i \rightarrow AP_j$, and y_{ij} represents its assigned channel.

B. SAP Selection Algorithm Procedure

The SAP selection algorithm finds the SAP set minimizing E by applying the routing tree and NIC/channel assignment algorithms in [7]-[9] to each feasible SAP set in the given AP network. Here, to reduce the computation time, this algorithm discards the undesirable SAP sets by adopting the *maximum estimated SAP load* before applying routing/assignment algorithms. The following procedure describes the details of the SAP selection algorithm:

(1) Calculate the lower bound N_{LB}^{SAP} on the number of SAPs, which is equal to the number of WDS clusters. If the input number of SAPs M is smaller than it, set M=$N_{LB}^{SAP} = \lceil \frac{N}{S} \rceil$.

(2) Generate a new SAP set by selecting M APs for SAPs among N APs.

(3) Check the feasibility of the SAP set in (2) by satisfying the three constraints. If it is not feasible, go to (8).

(4) Calculate the *maximum estimated SAP load* in Sect. II-C. If it is larger than the threshold there, go to (8).
(5) Apply the routing tree and NIC/channel assignment algorithms in [7]-[9]. If no feasible solution is obtained, go to (8).
(6) Calculate the cost function E.
(7) Update the best-found solution if E in (6) is smaller than the best one.
(8) Terminate the procedure if every SAP set is generated in (2). Otherwise, go to (2).

C. Maximum Estimated SAP Load

As the number of APs increases, the number of feasible SAP sets increases exponentially. Then, the computation time becomes unacceptably long, where the routing tree and NIC/channel assignment algorithms requires the inhibitory long time. To avoid this situation, the *maximum estimated SAP load* is calculated for each SAP set before their applications. If the value is larger than the threshold, it is discarded because traffics are concentrated into a specific SAP that may become the bottleneck of WIMNET.

1) SAP Selection Weight by Hop Count: As the cluster head to a CAP, a SAP with the smaller hop count (number of hops) has the larger possibility than a SAP with the larger hop count due to the delay. Thus, the *SAP selection weight* is calculated for any pair of a SAP and a CAP to represent the possibility of belonging to the same cluster. This weight for the hop count k is actually given by $W_k = 1/2^k$.

2) Procedure for Maximum Estimated SAP Load: For each SAP in a SAP selection, the *maximum estimated SAP load* is calculated by the following procedure:

(1) Find the connectable SAPs in the SAP set for each CAP.
(2) Calculate the weighted average of the *SAP selection weights* among the connectable SAPs for each CAP.
(3) Multiply the expected maximum number of associated hosts h_i to this weighted average for each CAP.
(4) For each SAP, sum up the values in (3) for all the CAPs that can select this SAP as the cluster head, which becomes the *estimated SAP load*.
(5) Select the maximum value of (4) among all the SAPs in the SAP set for the *maximum estimated SAP load*.

3) Example of Estimated SAP Load: Figure 2 illustrates an example of calculating the *estimated SAP load* for CAP-A. The three SAPs $\{1, 2, 3\}$ are connectable for CAP-A. Because the hop count from CAP-A is three, two, and three to each SAP, the corresponding *SAP selection weights* are given by: $1/2^3$, $1/2^2$, $1/2^3$. Because h_i for CAP-A is 10, the estimated load for each of the three SAPs by CAP-A is 2.5, 5, and 2.5 as shown in Figure. 2. Then, after calculating them for every CAP, the *estimated SAP load* is calculated by summing up them for each SAP.

4) Threshold for Traffic Concentration: The *maximum estimated SAP load* is compared with the given threshold Th to judge whether the SAP selection may cause the traffic concentration into a specific SAP or not. If it exceeds the threshold, the SAP selection is discarded. For this threshold, the twice of the average SAP load is used:

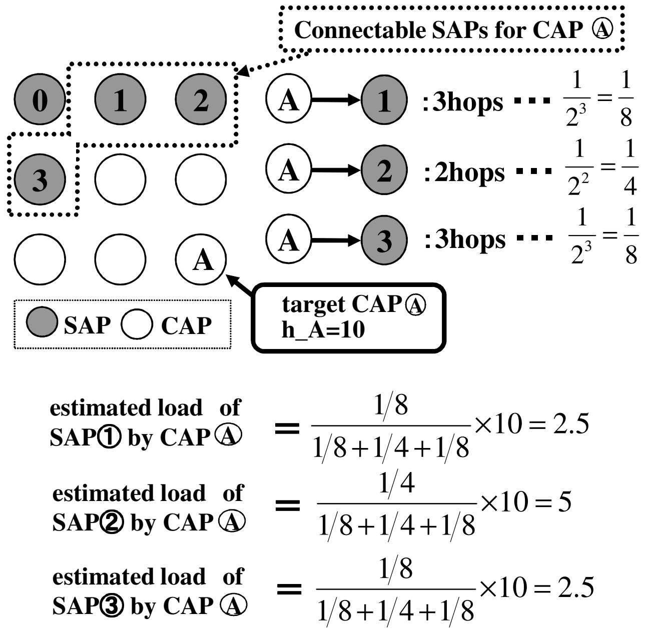

Fig. 2. Example of estimated SAP load calculation.

$$Th = \left(\frac{\sum_{i=1}^{N} h_i}{M} \right) \times 2. \tag{2}$$

III. Extension to Minimal SAP Selection for Maximal Throughput

In this section, we present an extension of the SAP selection algorithm to finding a minimal SAP set that provides the maximal throughput for WIMNET.

A. Motivation

The simulation results for the SAP selection algorithm in Sect. IV observed that the SAP set minimizing the cost function E provides the maximum throughput among possible SAP sets when the number of SAPs is fixed. Thus, if we find the SAP set giving the minimal value of this cost function by applying the SAP selection algorithm while increasing the number of SAPs starting from its lower bound, it can be a minimal SAP set for the maximal throughput. In our algorithm extension, the minimality of the cost function is determined by the sufficient decrease of the change of the cost functions between two consecutive numbers of SAPs. Actually, the minimality is determined when the ratio between two consecutive changes of the cost functions becomes smaller than a given parameter α ($\alpha = 0.5$ in this paper).

B. Algorithm Extension Procedure

The following procedure describes the details of this extension:

(1) Initialize the number of SAPs M by its lower bound: $M = N_{LB}^{SAP} = \lceil \frac{N}{S} \rceil$.
(2) Find the best SAP set that minimizes the cost function E for M, by applying the SAP selection algorithm in the previous section, and set $E_M = E$.
(3) Calculate the change of E by the increase of M, if $M \geq N_{LB}^{SAP} + 1$: $\Delta E_M = E_{M-1} - E_M$.

(4) Output the best SAP set with M SAPs, and terminate the procedure, if $M \geq N_{LB}^{SAP} + 1$ and $\Delta E_{M+1}/\Delta E_M < \alpha$.

(5) Otherwise, increment M by 1 and go to (3).

IV. Evaluation by Simulations

In this section, we evaluate the effectiveness of the proposed algorithms for the SAP selection through solving four instances.

A. WIMNET Simulator

To confirm the validity of our proposal, we evaluate the throughput of a solution found by our algorithm through packet transmission simulations using the WIMNET simulator [3]. In each simulation, every host posses 125 packets to the GW (Internet) and the GW does $1,000$ packets to every host before starting. Then, after every packet reached the destination, the throughput is calculated by dividing the total packet size with the simulation time. The bandwidth is set $30Mbps$ for any link between two APs and $20Mbps$ for that between an AP and a host. When multiple links within interference ranges try to be activated simultaneously, randomly selected one link among them is successfully activated, and the others are inserted into waiting queues.

B. Simulated Instances

As simulated instances, we adopt a grid topology of 25 APs with two traffic loads in Figures 3 and 4 for *instance 1* and *instance 2*. The center AP is selected as the GW. Each AP including the GW generally has wireless links with its four neighbors. Thus, the maximum of four NICs/channels can be assigned to the GW, which indicates that the largest possible throughput between the GW and its neighbors is $120Mbps$. The AP depicted by a gray circle is associated with 8 hosts for *instance 1* and 10 hosts for *instance 2*, whereas the AP by a white circle is associated with 1 host, so that nearly 100 hosts exist in any instance as the traffic load.

Then, for *instance 3* and *instance 4*, we adopt a hexagonal topology of 25 APs in Figures 5 and 6 that has been often used for cellular networks. The center AP is the GW. Each AP including the GW generally has wireless links with its six neighbors. Thus, the maximum of six NICs/channels can be assigned to the GW, which indicates that the largest possible throughput between the GW and its neighbors is $180Mbps$. The traffic patterns are the same as the above instances.

In any instance, the WDS cluster size S is set 8, the number of channels P is 8, and the maximum number of NICs per SAP Q is 8, so that any link incident to the GW can be assigned a different channel to maximize the throughput. In the following two subsections, we evaluate the effectiveness of the SAP selection algorithm for a given number of SAPs. Here, we allocate five SAPs for any instance from the lower bound on the number of clusters N_{LB}^{SAP} from $\lceil \frac{25}{8} \rceil = 4$. In each topology figure, an AP marked by a bold gray line represents a selected SAP, and a bold line between two APs represents a link in the routing tree by our algorithm.

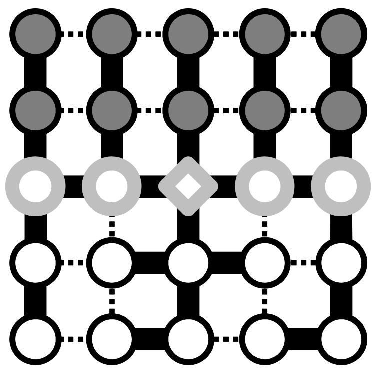

Fig. 3. Topology for *instance 1*.

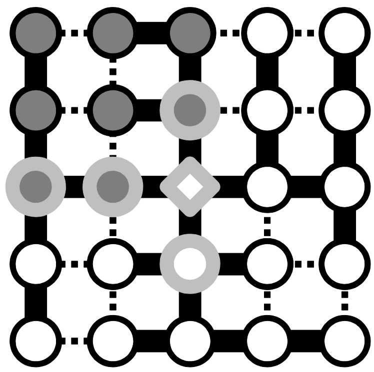

Fig. 4. Topology for *instance 2*.

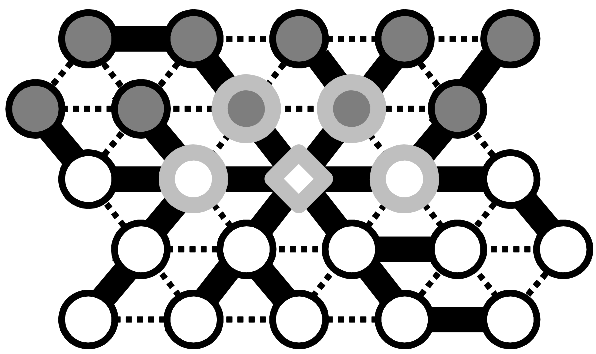

Fig. 5. Topology for *instance 3*.

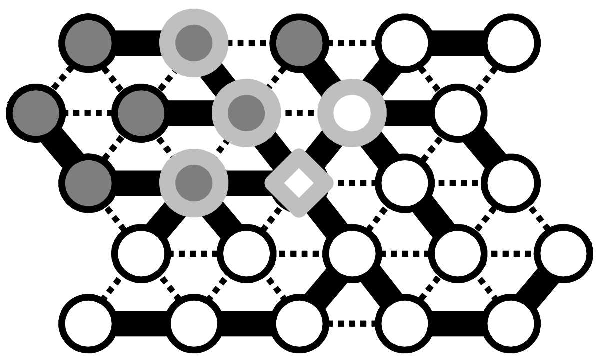

Fig. 6. Topology for *instance 4*.

C. Evaluation of Cost Function

First, we evaluate the validity of the cost function E_{traf} in terms of the throughput obtained by the WIMNET simulator. For each simulation, we use the output (the SAP selection, the routing tree and NIC/channel assignments) of our algorithm as the WIMNET configuration.

Figures 7-10 show the relationship between the cost function and the throughput for each instance respectively. In any result, the throughput becomes maximum when the cost function is minimum, which supports the validity of our cost function.

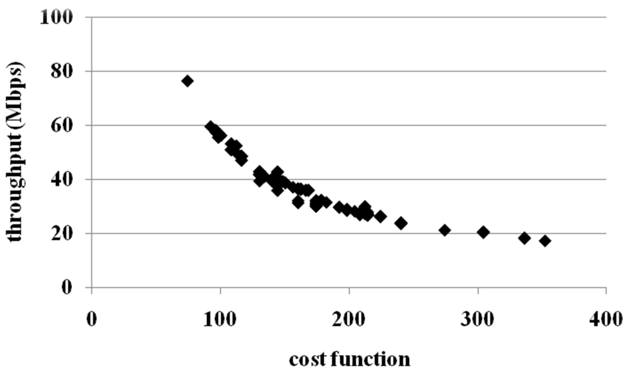

Fig. 7. Relationship between cost function and throughput for *instance 1*.

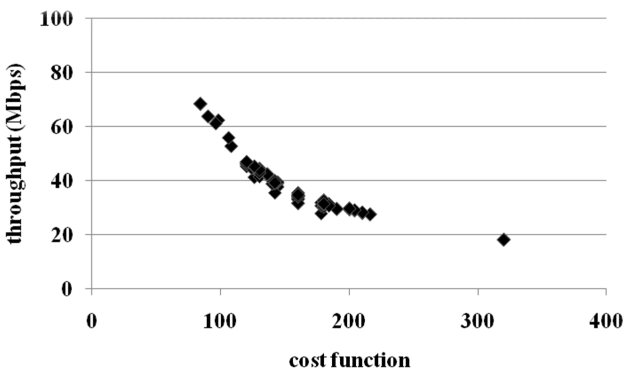

Fig. 8. Relationship between cost function and throughput for *instance 2*.

D. Evaluation of Maximum Estimated SAP Load

Then, we evaluate the effectiveness of the *maximum estimated SAP load* in reducing the computation time of our algorithm.

1) Validity of SAP Selection Weight: First, we verify the validity of the *SAP selection weight*. Table I shows the relationship between the hop count and the selected rate in the solutions such that a certain SAP/CAP pair is actually included in the routing tree by our algorithm. This table indicates that the selected rate roughly becomes $1/2$ every time the hop count increases by one, which supports the validity of our *SAP selection weight*.

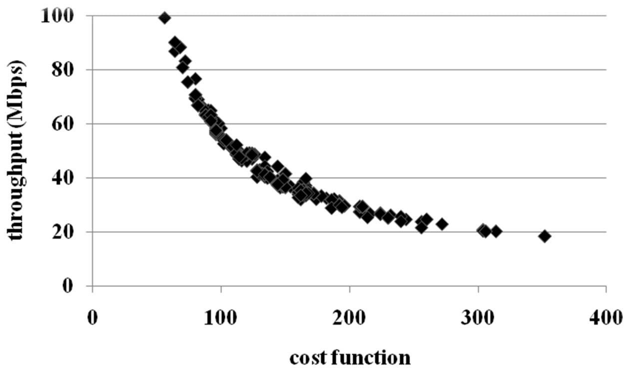

Fig. 9. Relationship between cost function and throughput for *instance 3*.

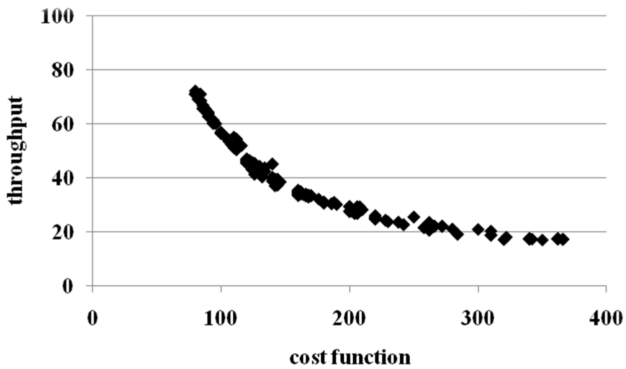

Fig. 10. Relationship between cost function and throughput for *instance 4*.

2) Effectiveness for Computation Time Reduction: Then, to evaluate the effectiveness of the *maximum estimated SAP load* in terms of the computation time reduction, we count the number of SAP sets where the routing tree and NIC/channel assignment algorithms are applied, when each of the three constrains ((1), (2), (3)) and the maximum estimated SAP load ((4)) is sequentially applied. Table II shows that our proposal reduces it into about 70% of the result by the constraints for *instances 1 & 2* and about 95% for *instances 3 & 4*.

TABLE I
RELATIONSHIP BETWEEN HOP COUNT AND SAP/CAP PAIR SELECTED RATE.

	1 hop	2 hops	3 hops	4 hops
instance 1	80.0%	38.5%	7.1%	0.0%
instance 2	72.7%	33.3%	22.2%	8.3%
instance 3	62.5%	34.6%	5.3%	0.0%
instance 4	62.5%	26.1%	19.0%	0.0%

TABLE II
NUMBER OF EXAMINED SAP SELECTIONS BY ALGORITHM.

	(1)	(1),(2)	(1) -(3)	(1)-(4)
instance 1	10626	243	243	184
instance 2	10626	243	243	176
instance 3	10626	490	490	475
instance 4	10626	490	490	458

TABLE III
ALGORITHM COMPUTATION TIME (SECONDS).

instance 1	*instance 2*	*instance 3*	*instance 4*
16.9	14.6	53.3	45.7

Table III shows the total computation time of our algorithm on a PC with *Core 2 Duo 2.4 GHz* for CPU, *8 GB* for main memory, and *FreeBSD 8.2-RELEASE* for OS. With the help of the *maximum estimated SAP load*, our algorithm can run within the acceptable time for WIMNET of 25 APs.

E. Evaluation of Algorithm Extension

Now, we evaluate the effectiveness of the algorithm extension to finding the minimal SAP selection for the maximal throughput. Figures 11-14 show the changes of the throughput (solid line) and the cost function (dotted line) in each of the four instances respectively, when the number of SAPs is increased one by one. In any instance, as the number of SAPs increases, the cost function decreases and the throughput increases, where they are saturated at a certain number of SAPs.

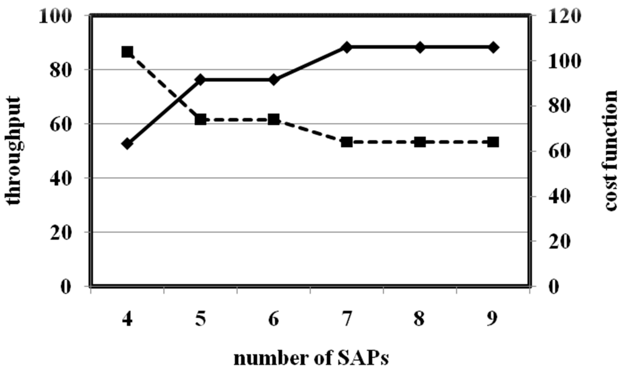

Fig. 11. Cost function and throughput by different number of SAPs for *instance 1*.

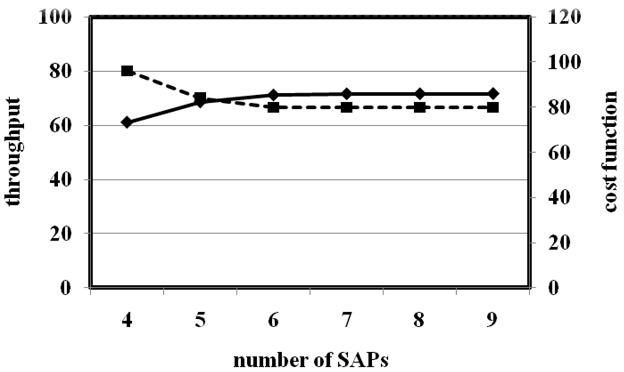

Fig. 12. Cost function and throughput by different number of SAPs for *instance 2*.

Table IV summarizes the number of SAPs in the solution by our algorithm extension and its throughput error to the maximum throughput for each instance. Note that the maximum throughput is given by selecting every AP as a SAP in each instance, and the error is calculated by $(b-a)/b \times 100\%$ where a is the throughput of the solution and b is the maximum one. In any instance, our algorithm extension can find a SAP set using only five SAPs for the near-maximum throughput. However, in *instance 1*, the throughput error is relatively larger than other instances. The investigation of this reason and its countermeasure will be in our future studies.

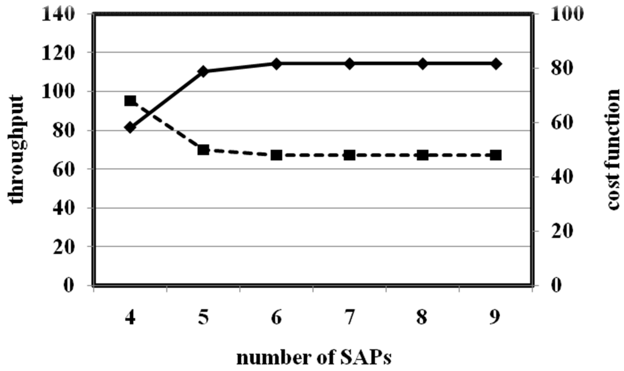

Fig. 13. Cost function and throughput by different number of SAPs for *instance 3*.

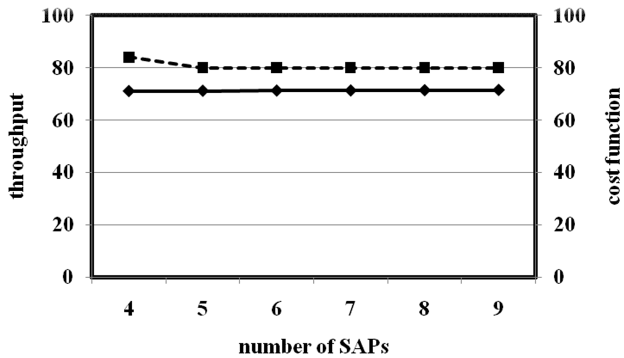

Fig. 14. Cost function and throughput by different number of SAPs for *instance 4*.

V. CONCLUSION

In this paper, we first presented the smart access-point (SAP) selection algorithm to find an optimal hierarchical structure for the wireless Internet-access mesh network *WIMNET*. Then, we extended this algorithm to finding the minimal SAP set for the maximal throughput. We verified the effectiveness of our proposals through simulations in four instances using the WIMNET simulator. Our future works may include the improvement of the algorithm extension to the minimal SAP selection, the SAP selection algorithm under dynamic changes of traffic loads and/or network topology by failures of links/APs, and their evaluations in real networks after implementing the SAP.

REFERENCES

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Comput. Network*, vol. 47, no. 4, pp. 445-487, March 2005.

TABLE IV
SOLUTION QUALITY OF SAP SET SELECTED BY ALGORITHM.

topology	algorithm			maximum	
	# of SAPs	throughput(Mbps)	error(%)	# of SAPs	throughput(Mbps)
instance 1	5	76.5	13.5	25	88.5
instance 2	5	68.6	5.1	25	72.3
instance 3	5	110.4	3.5	25	114.4
instance 4	5	71.1	1.5	25	72.2

[2] Y. Zhang, J. Luo, and H. Hu ed., *Wireless mesh networking: architectures, protocols and standards*, Auerbach Pub., New York, 2007.
[3] N. Funabiki ed., *Wireless mesh networks*, InTech, Jan. 2011, online available at http://www.intechopen.com/books/show/title /wireless-mesh-networks.
[4] 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. 784-739, March 2009.
[5] S. Tajima, N. Funabiki, and T. Higashino, T., "A WDS clustering algorithm for wireless mesh networks", *IEICE Trans. Inform. Systems*, vol. E93-D, no. 4, pp. 800-810, April 2010.
[6] K. Hirakata, T. Horiuchi, N. Funabiki, and T. Nakanishi, "A construction of the smart access point for practical wireless mesh networks," *IEICE Tech. Report*, vol. NS2008-99, pp. 63-68, Nov. 2008.
[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-2007)*, pp. 240-245, Nov. 2007.
[8] N. Funabiki, K. Uemura, T. Nakanishi, and W. Hassan, "A minimum-delay routing tree algorithm for access-point communications in wireless mesh networks," *Proc. Int. Conf. Research Innovation Vision for the Future (RIVF-2008)*, pp. 161-166, July 2008.
[9] K. Uemura, N. Funabiki, and T. Nakanishi, "A communication route optimization algorithm for scalable wireless mesh networks," *IEICE Trans. (B)*, vol. J92-B, no. 9, pp. 1526-1537, Sep. 2009.
[10] K. Uemura, N. Funabiki, and T. Nakanishi, "A proposal of a smart access point allocation algorithm for scalable wireless mesh networks," in *Lecture Notes in Engineering and Computer Science: Proc. International MultiConference of Engineers and Computer Scientists (IMECS 2010)*, 17-19 March, 2010, Hong Kong, pp. 848-853.
[11] N. Funabiki, T. Takebayashi, and T. Nakanishi, "Minimal smart access-point selection for maximal throughput in wireless mesh networks," in *Lecture Notes in Engineering and Computer Science: Proc. International MultiConference of Engineers and Computer Scientists (IMECS 2011)*, 16-18 March, 2011, Hong Kong, pp. 594-599.
[12] S. Sukaridhoto, N. Funabiki, and T. Nakanishi, "A proposal of a traffic control method with consumed bandwidth estimation for real-time applications in wireless mesh networks," *Proc. IEEE Symp. Consumer Electronics (ISCE 2011)*, June 2011.
[13] N. Funabiki, J. Shimizu, T. Nakanishi, and K. Watanabe, "A proposal of an active access-point selection algorithm in wireless mesh networks," *Proc. Int. Conf. Network-Based Information Systems (NBiS 2011)*, Sep. 2011.
[14] S. Lakshmanan, K. Sundaresan, and R. Sivakumar, "On multi-gateway association in wireless mesh networks," *Proc. IEEE Workshop. Wireless Mesh Networks (WiMesh)*, pp.64-73, Sep. 2006.
[15] A. Raniwala, K. Gppalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM Mobile Comput. Commun. Review*, vol. 8, no. 2, pp. 50-65, 2004.
[16] A. Raniwala, K. Gppalan, and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," *Proc. IEEE Infocom*, vol. 3, pp. 2223-2234, 2005.

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 (currently, Department of Electrical and Communication Engineering) at Okayama University as a professor. His research interests include computer networks, optimization algorithms, educational technology, Web technology, and network security. He is a member of IEEE, IEICE, and IPSJ.

Sritrusta Sukaridhoto received the B.S. degree in Electrical Engineering from Institute Technology Sepuluh November of Surabaya, Indonesia, in 2002. In 2002, he joined the Department of Telecommunication Technology at Electrical Engineering Polytechnic Institute of Surabaya, Indonesia, as a lecturer. He is currently a Ph.D. candidate in Graduate School of Natural Science and Technology at Okayama University, Japan. His research interests include computer networks. He is a student member of IEEE and IEICE.

Masaharu Hata received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyushu University in 1973, 1975, and 1983, respectively. He joined the Electrical Communication Laboratories, NTT in 1975. He engaged in research and development of digital cellular systems and their international standardization. From 1992 to 2001, he was with NTT DoCoMo, Inc., where he was the director of System Technology Laboratory in Wireless Laboratories. Since 2001, he has been with Okayama University, where he is a professor in Graduate School of Natural Science and Technology. He is a fellow of IEICE and a senior member of IEEE.

Shigeru Tomisato received the B.S. and Ph.D. degrees in communication engineering from Osaka University, Japan, in 1987 and 2002, respectively. He joined Nippon Telegraph and Telephone Corporation (NTT), Japan, in 1987, and was with NTT DoCoMo, Inc., Japan, from 1998 to 2003. In 2003, he joined the Department of Communication Network Engineering at Okayama University, Japan, as an associate professor. His research interests include adaptive signal processing technologies for mobile radio communications such as adaptive arrays, MIMO channel signal transmission, and OFDM transmission. He is a member of IEEE and IEICE.

Toru Nakanishi received the M.S. and Ph.D. degrees in information and computer sciences from Osaka University, Japan, in 1995 and 2000, respectively. He joined the Department of Information Technology at Okayama University, Japan, as a research associate in 1998, and moved to the Department of Communication Network Engineering in 2000, where he became an assistant professor in 2003 and an associate professor in 2006, respectively. His research interests include cryptography, information security, and network protocol. He is a member of IEICE and IPSJ.

Kan Watanabe received the B.S., M.S., and Ph.D. degrees in information technology from Tohoku University, Japan, in 2006, 2008, and 2011, respectively. In 2011, he joined the Department of Electrical and Communication Engineering at Okayama University, Japan, as an assistant professor. His research interests include distributed systems and wireless networks. He is a member of IEEE, IEICE, and IPSJ.

Shigeto Tajima received the B.S. degree in industrial engineering from Osaka Electro-Communication University, Japan, in 1992 and the Ph.D. degree in information and computer sciences from Osaka University, Japan, in 2010. 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 assistant professor in 1996. His research interests include computer networks and optimization algorithms. He is a member of IEICE and IPSJ.