

Cross-layer selective routing for cost and delay minimization in IEEE 802.11ac wireless mesh network

I.-Wei Lai¹ · Nobuo Funabiki² · Shigeto Tajima³ · Md. Selim Al Mamun² · Sho Fujita²

Published online: 14 February 2017
© Springer Science+Business Media New York 2017

Abstract A Wireless Internet-access Mesh NETWORK (WIMNET) provides scalable and reliable internet access through the deployment of multiple access points (APs) and gateways (GWs). In this work, we propose a selective routing algorithm aiming at a *hierarchical minimization* of the operational cost and the maximal end-to-end delay. In particular, by deploying redundant APs/GWs in the network field, the WIMNET becomes robust to the link or AP/GW failure. However, these redundant APs/GWs increase the operational cost like the power consumption. By using Dijkstra algorithm and 2-opt algorithm, the proposed algorithm iteratively deactivates the deployed APs/GWs and performs the routing that reduces the maximal end-to-end delay based on the APs/GWs remaining active. The generated route meets the real-world constraints like fairness criterion. We further propose a cross-layer design to enhance the routing performance by exploiting the MAC-layer frame aggregation technique. The selective routing algorithm is then implemented in the WIMNET simulator proposed by our group. The numerical experiments demonstrate that in both indoor and open space environments, the proposed selective routing greatly reduces the operational cost, i.e., up to 80% APs/GWs can be deactivated.

Keywords A wireless mesh network · IEEE 802.11ac · Routing · Multi-hop communication · Cross-layer design · Hierarchical optimization

1 Introduction

The IEEE 802.11 wireless local area network (WLAN) has been deployed all over the world [1, 2]. Compared with the wired network, the wireless connections between the access points (APs), gateways (GWs),¹ and the hosts make the WLAN more flexible, scalable, and inexpensive. For a wider area or more users, multiple APs are distributed to increase the coverage and the capacity. Among several variations, this work focuses on a simple *Wireless Internet-access Mesh NETWORK (WIMNET)* [3–5] scheme that uses APs as wireless mesh routers and linked to the Internet through the GWs. The host either directly connects to the GW, or is relayed by the AP in the close vicinity, leading to a multi-hop relay network, as illustrated in Fig. 1.

In practice, the AP/GW may be corrupted due to the hardware or software faults, and the wireless link may be disconnected due to the change of the environment, which are respectively termed as the node and link failures. When the size of WIMNET increases, such failures appear more frequently and impair the communication performance. For example, a node failure of an AP causes all the associated hosts to be disconnected from the networks. To solve this problem, redundant APs/GWs are deployed in the network field, so as to back up the failures [6]. Nonetheless, activating these redundant APs/GWs introduces extra power

✉ Md. Selim Al Mamun
mamun0013@s.okayama-u.ac.jp

¹ Department of Electrical Engineering, National Taiwan Normal University (NTNU), Taipei, Taiwan

² Department of Electrical and Communication Engineering, Okayama University, Okayama, Japan

³ Graduate School of Engineering Science, Osaka University, Toyonaka, Japan

¹ A GW can be considered as an AP with the wired connection to the Internet.

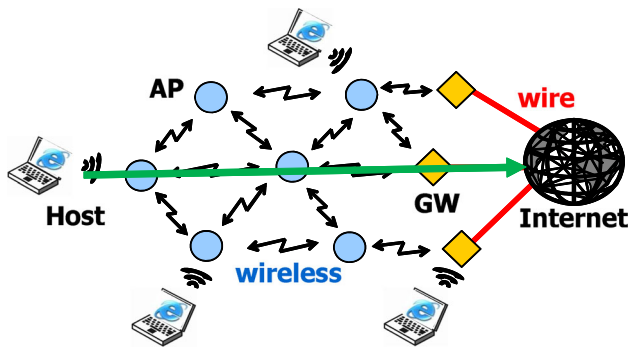


Fig. 1 Illustration of WIMNET

consumptions, which is reflected as the network operational cost [7].

Several researches are conducted to reduce the operational cost of the network. For example, in [9], a distributed power control is proposed to improve the energy efficiency of routing by estimating the amount of power that is needed for reliable communications over any link in the network. In [8], the active AP selection algorithm is introduced to strike a good balance among the network reliability, the operational cost, and throughput. Given a network setting and constraint, such algorithm can meet the system requirement by only activating parts of the APs deployed in the network. By formulating the active AP selection as a combinatorial optimization, a heuristic approach is designed to reduce the number of active APs.

While conventionally the routing algorithm are designed to optimize the communication performances, e.g., delay, interferences, packet loss, and load at GW [10, 12–14], in this work, we propose a *selective routing* algorithm to jointly solve the routing and AP selection problems in a practical IEEE 802.11ac WIMNET. In particular, we formulate a hierarchical optimization problem, which first reduces the number of active APs/GWs and then lowers the maximum end-to-end delay. The maximal end-to-end delay is selected as the communication performance metric since it is crucial for many applications such as video LIVE broadcasting and cloud service, which also is termed as the *interaction delay* [17].

The proposed selective routing iteratively deactivates the APs/GWs and generates a feasible routing that reduces the maximum end-to-end delay under several practical constraints, e.g., the fairness criterion to guarantee the minimum throughput of each host. In addition to these, we consider the constraints raised by the implementation of the IEEE 802.11ac including the dynamic link speed and the hop limitation. For the former, our measurement results show that link speed of the IEEE 802.11ac commercial products varies rapidly with respect to the transmission distance; for the latter, the available IEEE 802.11ac

commercial products limit their number of hops to two. This is because that, due to the proportional end-to-end delay with respect to the number of hops in the relay transmission, the larger number of hops, the higher packet loss probability and hence poorer transmission reliability.

We then leverage the *cross-layer* design of the MAC layer and network layer to further enhance the routing performance [11–13]. Specifically, the *frame aggregation* is a MAC-layer technique used to improve the link speed by removing the frame overhead. We consider such link speed enhancement into our selective routing algorithm so as to further reduce the number of active APs/GWs and the maximal end-to-end delay. Numerical example shows that, up to 80% APs/GWs can be deactivated with slight additional end-to-end delay. Last but not least, we discover that realizing such constraint selective routing may be time-consuming in some cases. Consequently, the *plural AP deactivation* is proposed to simultaneously deactivate multiple APs, which results in half CPU time saving reported by the numerical examples. In short, the features of the proposed selective routing algorithm are listed below:

- ① The constraint hierarchical minimization including the number of active AP minimization and the maximum end-to-end delay minimization,
- ② The considerations of practical IEEE 802.11 ac WIMNET constraints: multiple GWs, fairness, dynamic link speed, and hop limitation,
- ③ The cross-layer design that exploits frame aggregation,
- ④ The plural AP deactivation to speed up the selective routing algorithm.

The rest of this paper is organized as follows: Sect. 2 shows the system model, optimization formulation, the IEEE 802.11 ac link speed measurement result, and its approximate formula. Section 3 elaborates on the flow of the selective routing algorithm and the minimization of the active APs. Section 4 introduces the details of the maximal end-to-end delay minimization. The WIMNET simulator which integrates the IEEE 802.11ac features for this selective routing design is explained in Sect. 5. Section 6 provides numerical simulation results to demonstrate the effectiveness of the proposed algorithms. Finally, conclusions are drawn in Sect. 7.

2 System model

The topology of the WIMNET is illustrated in Fig. 1, where in this work we denote the *node* as a host, an AP, or a GW, the *link* as a connection between two nodes, and

the *path* as an end-to-end connection between a host and a GW. The links and paths are generated by the routing algorithm, which optimizes certain communication performance metric, e.g., delay, interferences, packet loss, and load at GW [15, 16]. We consider commercial IEEE 802.11ac products, which the route can only be established with the relay path comprising less than or equal to two links, i.e., the hop limitation. In a WIMNET, the AP node or link may fail due to the software faults, hardware faults, or changes of the environment. When a failure occurs, multiple hosts associated with this node or link are disconnected. For example, the relay paths with a failed AP is cut off, and thus hosts along this relay path are disconnected from the wireless networks. To remedy the performance degradations caused by such disconnection, redundant APs are deployed in the network field [6].

Although this redundancy enhances the network reliability, activating more APs introduce the power consumptions [7, 8], which implies that one should reduce the number of active APs when the operational cost is essential. We thus propose the *selective routing* algorithm to simultaneously reduce the number of active APs and the communication performance. In other words, the selective routing optimizes communication performance by using minimal active APs. If the node/link failure is sensed, the selective routing is executed again to generate a new route which excludes the failed APs. In this case, we can strike a good balance between the network reliability and the extra cost introduced by the redundant APs.

The inputs of the selective routing are the network topology including the node set, node type (GW, AP, host), node location coordinate, link set, the link speed formula which will be expounded later in Sect. 2.2, and the link speed threshold for routing. The speed threshold is used to describe the AP/GW coverage, i.e., the link having speed below this threshold is regarded as absent. With these inputs, the selective routing algorithm generates the routes by only utilizing a small fraction of the deployed APs/GWs. Certain communication performance metric is optimized as well. In this work, the maximal end-to-end delay is selected as the communication performance metric for the optimization as an example. Different communication metrics can be optimized through the optimization formulation introduced in the following subsections. Additionally, to make the proposed selective routing more practical, a measurement campaign is conducted to evaluate the dynamic link speeds for IEEE 802.11ac WLAN. In Sect. 2.2, we explain the campaign and show an approximate model for the IEEE 802.11ac WLAN based on the measurement results.

2.1 Hierarchical minimization of active AP and maximal end-to-end latency

The selective routing algorithm jointly reduces the number of active APs/GWs and the maximal end-to-end delay, which can be formulated as a multi-objective optimization [18]:

$$\underset{\mathcal{R}(\mathbf{a})}{\text{minimize}} \left\{ N_A, \max_{k \in \mathcal{H}} \tau_k \right\}, \quad (1)$$

where N_A is the number of active APs, τ_k is the end-to-end delay of the k th host, and \mathcal{H} denotes the set of the hosts. $\mathbf{a} \in \{0, 1\}^N$ is a binary vector with length N equal to the number of APs and GWs deployed in the network. We call this vector as *AP/GW indicator* since it represents the active state of the AP/GW: “0” and “1” respectively indicate inactive and active. For example, when the i th entry is zero, i.e., $a_i = 0$, the associated i th AP is deactivated. Thus, we have the relationship

$$\|\mathbf{a}\|_1 = N_A \quad (2)$$

between the number of activate APs and the AP indicator vector, where $\|\cdot\|_1$ denotes the l^1 -norm (Hamming weight) of a vector. Aiming at solving (1), the selective routing generates the route $\mathcal{R}^*(\mathbf{a}^*)$ with the selection of the active APs \mathbf{a}^* that yields a low value of $\|\mathbf{a}\|_1$ as possible.

However, for the multi-objective optimizations, there are generally multiple *Pareto optimal solutions*. A Pareto solution in our case means that one cannot minimize the number of active APs without increasing the maximal end-to-end delay, and vice versa. Since deciding a preferred point from a set of Pareto optimal solutions depends on the system preference and is not the main focus on this work, we study one of the instances of the multi-objective optimization, where the active AP minimization has the higher priority. Given the number of active APs, the maximal end-to-end delay is then reduced. In this way, (1) is formulated as an *hierarchical minimization* [19]:

$$\begin{aligned} &\underset{\mathcal{R}(\mathbf{a}^*)}{\text{minimize}} \quad \max_{k \in \mathcal{H}} \tau_k \\ &\text{subject to} \quad N_A^* = \min_{\mathbf{a}=\{0,1\}^N} N_A, \end{aligned} \quad (3)$$

where the objective function (3) is the min–max end-to-end delay, under the constraint of minimal number of active APs. This implies that, N_A and the associated AP indicator \mathbf{a} should be designed first. With the given \mathbf{a}^* , the routing $\mathcal{R}^*(\mathbf{a}^*)$ is then generated to reduce the maximum end-to-end delay.

For the end-to-end delay, since multiple hosts may be connected to an AP or a GW, the end-to-end delay depends on the multiple access protocol and the scheduling. In this

work, we assume that all links rooted from the same GW share a single frequency channel, and different GWs use different frequency channels. In this case, the k th host may suffer from a long end-to-end delay when the traffic is huge so that the packets have to be buffered in the AP due to the spectrum unavailability. For the k th host rooted from the m th GW, its maximal end-to-end delay τ_k is upper bounded by d_m , the accumulated delays of all links rooted from the m th GW:

$$d_m = \sum_{k \in \mathcal{H}GW(m)} \left(\sum_{(i,j) \in \mathcal{P}_k} \frac{r_k}{s_{ij}} \right), m \in \mathcal{G} \quad (4)$$

where \mathcal{P}_k denotes links ij of the node i and j in the relay path associated with the k th host, $\mathcal{H}GW(m)$ indicates the set of hosts connected to the m th GW. \mathcal{G} is the GW set. For the parameters associated with the inner summation, s_{ij} is the link speed between the node i and j , and r_k indicates the traffic volume of the k th host.² By using

$$\max_{k \in \mathcal{H}} \tau_k \leq \max_{m \in \mathcal{G}} d_m \quad (5)$$

and (3), we formulate the optimization that the proposed selective routing algorithm solves:

$$\begin{aligned} & \underset{\mathcal{R}(\mathbf{a}^*)}{\text{minimize}} && \max_{m \in \mathcal{G}} d_m \\ & \text{subject to} && N_A^* = \min_{\mathbf{a} = \{0,1\}^N} N_A, \end{aligned} \quad (6)$$

with d_m defined in (4). In addition to the AP minimization constraint, other constraints related to the routing are enumerated as follows:

- *Node type connection* For any node, either of the following nodes must be selected for the next node, starting from a GW.
 - For GW: AP or host,
 - For AP: Host, and
 - For host: None.
- *Unique routing* For any node, only one upstream node toward a GW must exist.
- *Multiple GWs* The routing should support WIMNET with multiple GWs.
- *Host covering* Every host must be connected with a GW in the routing forest.
- *Hop limitation* The number of hops between a GW and a host is less than or equal to 2.
- *Fairness* Each host should be able to have throughput larger than a certain value.

² The GW can sense and track the traffic volume of each host and update this value by the on-line learning process.

For the last constraint, since the host throughput depends on many factors as the end-to-end delay in (4), we approximate the k th host throughput $\hat{s}_{H,k}$ by the following equation

$$\hat{s}_{H,k} = \frac{1}{\sum_{(i,j) \in \mathcal{P}_k} \frac{1}{s_{ij}}}, \quad (7)$$

where $\frac{1}{s_{ij}}$ represents the link delay to send 1 bit through the link ij with link speed s_{ij} . Thus, $\sum_{(i,j) \in \mathcal{P}_k} \frac{1}{s_{ij}}$ can be interpreted as estimated latency to send 1 bit through the links connecting the GW and the k th host in the relay path. Consequently, the inverse of the estimated latency becomes the estimated throughput of the k th host. If this estimated throughput for any host is below the threshold sT , i.e., the guaranteed minimum host throughput, the routing result is regarded as infeasible. Fulfilling all the constraints mentioned above, the selective routing generate a set of routing trees, namely, the routing forest, where the roots and leaves are respectively served by GWs and hosts.

2.2 Throughput measurement of IEEE 802.11ac

From the previous sections, we know that the link speed affect the objective function d_m and also the fairness criterion, which is an important parameter for routing. The link speed is affected by many factors like modulation and coding schemes, transmission power, transmission distance, and even the receiver design [20]. Therefore, the theoretical computation of the link speed is difficult. In this work, we take an alternative which conducts a measurement campaign to evaluate the actual link speed. Since the modulation, coding scheme, and the transmission power are optimally adjusted by the commercial device, the link speed can be solely characterized by the adopted device and the transmission distance, under the assumption of same environment and thus similar channel profile. We thus can establish a link speed formula with transmission distance as the single input based on the measurement results [21] by applying the regression analysis. This link speed formula greatly simplify the optimization complexity.

For the measurement campaign, two personal computers with the setting listed in Table 1 are prepared respectively as the source and destination nodes of an IEEE 802.11ac transmission. Figure 2 shows the measurement results for the transmission distance ranging from 0 to 100m. The parameters in *iperf* are set as: 50 sec for the measurement time, 477 kbytes for the window size, and 8 kbytes the buffer size. We can see that the link speed drastically slumps with the increase of the transmission distance. To reflect such speed variation, we apply the regression analysis to yield the approximate formulate $f(x)$:

Table 1 Devices and softwares for the measurement of the IEEE 802.11ac link speed

PC1	Model	Ultrabook Lesance NB S3441/L
	CPU	Intel (R) Core i5 3317U (2.6 GHz)
	OS	Windows7
PC2	Model	Ultrabook Lesance NB S3532-SP
	CPU	Intel (R) Core i3 2350M (2.3 GHz)
	OS	Windows7
NIC		Buffalo WZR-G1750DHP
Software		iperf2.0.5
Protocol		TCP

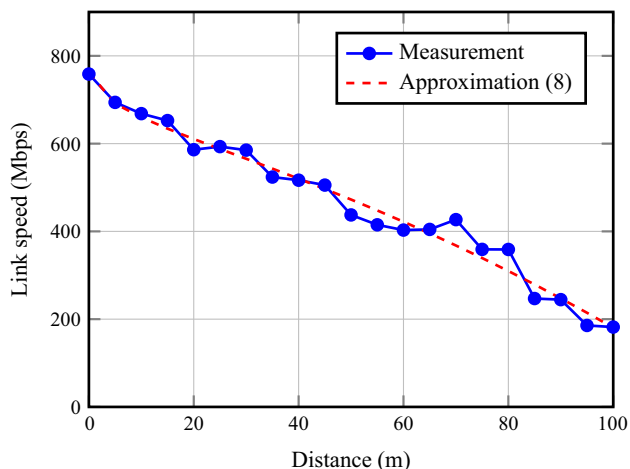


Fig. 2 Dynamic IEEE 802.11ac link speed measurement results

$$f(x) = 762.4 - 31.58\sqrt{x} - 2.66 \cdot 10^{-2}x^2, \tag{8}$$

where the square root and quadratic terms, i.e., \sqrt{x} and x^2 , are heuristically determined for better accuracy when three terms are used in the right-hand side. Such formula is used to develop the selective routing algorithm as will be expounded later.

3 Selective routing for hierarchical optimization

In this section, we propose the selective routing algorithm to iteratively solve the hierarchical optimization in (6), which first aims at minimizing the number of active APs/GWs, and then reduces the maximal end-to-end delay with a given set of activated APs/GWs. As depicted in Fig. 3, the proposed algorithm comprises the preprocessing module, the iterative module, and the postprocessing module. The preprocessing module is used to speed up the algorithm by simultaneously deactivating multiple APs. The iterative module sequentially select an

AP/GW as a candidate for the hypothesis of deactivation. In particular, whenever a candidate AP/GW is selected, the *min-max-delay routing* is executed to check if a feasible route can be generated without using the candidate AP/GW. If so, this AP is deactivated. Otherwise, the algorithm returns to the state of candidate AP/GW selection to choose another AP/GW as the next candidate. The min-max-delay routing aims at minimizing the maximal end-to-end delay, which will be expounded more in Sect. 4. After all the APs/GWs are examined, the iterative module terminates and the postprocessing starts to refine the minimization of active APs/GWs. In the following subsections, we respectively elaborate on each module.

3.1 Preprocessing: plural AP deactivation

In this preprocessing module, multiple APs are deactivated to reduce the run time of the selective routing. Specifically, the two-hop constrain means that, the communication path between the host and GW comprises at most one AP. Since the link speed linearly decreases as the transmission distance increases as depicted in Fig. 2, the APs located around the middle between a GW and a host are more likely to provide high host throughput. Thus, we deactivate the APs near GWs or near hosts are deactivated according to the information of geometric locations. The details of the preprocessing is given as below:

Plural AP Deactivation

- ① Calculate the distance between any pair of a GW and a host
 - ② Select the largest distance computed in ① and define it as L
 - ③ Select APs whose distances from half of the GWs are smaller than $1 / 3L$ or larger than $2 / 3L$
 - ④ Deactivate the AP selected in ③ one at a time. If deactivating this AP results in infeasible routing, skip this deactivation and select another AP
 - ⑤ If all the selected APs are examined, terminated the step
-

3.2 Iterative module: AP/GW selection and deactivation

The iterative module is the main body of the selective routing, which chooses a candidate AP and evaluates if a feasible route can be generated without using this candidate AP. To introduce this iterative module, we first define four types of APs: *active AP*, *inactive AP*, *candidate AP* and *must-active AP*. Initially, the APs deactivated by the preprocessing module are categorized into the inactive AP. The remaining APs belong to the set of

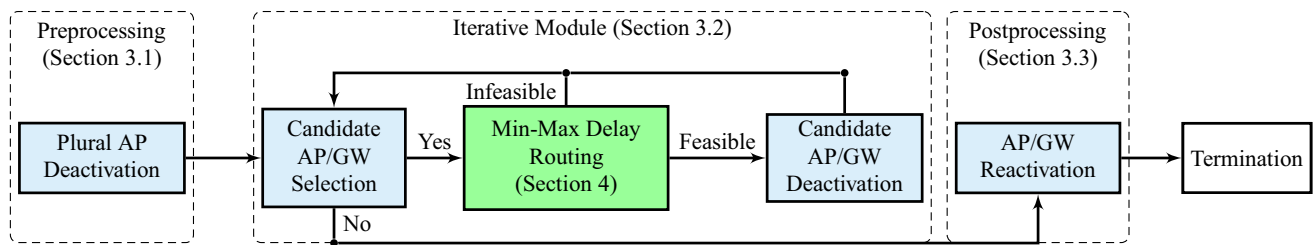


Fig. 3 Flow chart of the proposed selective routing algorithm

active APs. When the iterative module starts, we select the active AP with minimum number of connected hosts as the candidate AP. This selection criterion tries to *reduce the influence* to the routing. Then, without using this candidate AP, the *min-max-delay routing* is performed to reduce the maximal end-to-end delay. The details of the min-max-delay routing is elaborated on more in Sect. 4. If a feasible routing can be established without using this candidate AP, this candidate AP is deactivated and assigned to the inactive AP set. Otherwise, this candidate AP is categorized as a must-active AP. The above-mentioned procedure can be viewed as an approach that iteratively nulls the entries of \mathbf{a} in (6) so as to reduce N_A . The min-max-delay routing is applied to generate the route conditioned on a specific \mathbf{a} . This iterative module repeats until the set of active AP becomes empty such that no candidate AP can be selected.

After the active AP set becomes empty, we start to deactivate the APs that serve as GWs. The GW deactivation process resembles that of the AP deactivation, where *active GW*, *inactive GW*, *candidate GW* and *must-active GW* are specified. Then, the active GWs are sequentially selected as a candidate GW, which will be classified to the inactive GW or must-active GW sets, depending on the routing result. When both active AP and GW sets are empty, implying that no more candidates for deactivation can be selected, the iterative module terminates.

3.3 Postprocessing: AP/GW reactivation

After sequentially deactivating APs and GWs such that no candidate AP/GW can be selected from the active AP/GW set, the AP/GW reactivation starts to further refine the minimization result by using the local search. Specifically, we repeatedly reactivate a deactivated AP or GW from the inactive AP/GW set and examine if those must-active APs and GWs can be deactivated after this reactivation. When activating such AP/GW allows more than one must-active

APs or GWs to be deactivated, a better local minimum is approached. The details of the AP/GW reactivation is explained below:

AP/GW Reactivation

- ① Make a list of the inactive APs and GWs by sorting them in the deactivated order, including those deactivated in this AP/GW reactivation procedure. Initialize the reactivation flag $FR = 0$.
- ② Reactive the first AP/GW in the list. If more than one must-active APs or GWs can be deactivated, i.e., a feasible routing can be established with less active APs and GWs, deactivate them and set $FR = 1$. Otherwise, resume the previous routing result and active AP/GW selection.
- ③ Remove the reactivated AP or GW from the list. If the list is non-empty, go to ②.
- ④ If $FR = 1$, go to ①. Otherwise, terminate the reactivation process.

4 Min-Max delay routing

In this section, we explain the min-max-delay routing that generates a routing forest under a specific set of active APs. This routing has to meet other constraints mentioned in Sect. 2, e.g., multiple GWs and the fairness criterion. Moreover, to enhance the performance, we introduce the cross-layer design by exploiting the frame aggregation technique of IEEE 802.11ac. This routing method seeks a routing forest that reduces the maximum end-to-end delay through six steps as delineated in Fig. 4, where the first two steps are used to generate the routing forest, and the rest are used to refine the routing to reduce the maximum end-to-end delay. In particular, the fifth step exploits the frame aggregation.

In the first step, the *routing graph* is generated to enumerate links that can be adopted by the routing, which generates a routing forest based on active APs indicating by $\hat{\mathbf{a}}$.

Step 1 (Routing Graph Generation)

- ① Every GW, AP, and host is selected as a vertex in the routing graph.
 - ② The following pair of vertices is connected by an edge in the routing graph as a possible link in the routing forest:
 - A GW and an AP,
 - A GW and a host, and
 - An AP and a host,
 if the corresponding link speed is equal to or larger than the link speed threshold.
-

In the second step, we initialize the route $\mathcal{R}(\hat{\mathbf{a}})$ conditioned on a set of active APs by applying the Dijkstra algorithm [22]. Specifically, letting the edge weight of a link be the inverse of the corresponding link speed, for a pair of source and destination nodes in a graph, the Dijkstra algorithm finds the shortest path between them, i.e., the one with minimum accumulated edge weight.

Step 2 (Initial Route Generation)

- ① For each host, find the shortest path to each GW in the routing graph under the constraints mentioned previously.
 - ② Given a host, select its shortest path among those in ①. Add this path to F .
 - ③ Remove the selected host in ② and the incident links from the routing graph.
 - ④ If the path in ② contains an AP, remove the links between this AP and other GWs from the routing graph.
 - ⑤ If no host exists in the routing graph, calculate the maximum end-to-end delay d_m defined in (4) and terminate the procedure. Otherwise, go to ①
-

The following steps are used to reduce the maximal end-to-end delay of $\mathcal{R}(\hat{\mathbf{a}})$. In the third step, we identify the bottleneck, i.e., the m th GW with maximal end-to-end delay d_m in (4), and reduce this delay by moving the APs in this routing tree to other routing trees.

Step 3 (AP Path Improvement)

- ① Select the routing tree rooted at the m th GW with the maximum end-to-end delay d_m and term it as the *bottleneck tree*.
 - ② Reduce d_m by moving the APs in the bottleneck tree to other routing tree, that is, connect to other GWs:
 - 1) Select one AP in the bottleneck tree.
 - 2) Connect this AP to other GWs and compute the new maximum end-to-end delay. If this new maximum end-to-end delay is reduced, confirm this new connection.
 - 3) Go to 2) until all the GWs (except the m th GW) are examined.
 - 4) If all the APs in the bottleneck tree are examined, terminate this step. Otherwise, go to 1).
-

The fourth step resembles to the third step. Specifically, we identify the bottleneck GW and reduce the maximal end-to-end delay by moving the hosts in this routing tree to other routing trees.

Step 4 (Host Path Improvement)

- ① Select the bottleneck tree.
 - ② Improve the routing forest by moving the hosts in the bottleneck tree to other routing tree, that is, connect to other GWs:
 - 1) Select one host in the bottleneck tree.
 - 2) Connect this host to other GWs and compute the new maximum end-to-end delay. If this new maximum end-to-end delay is reduced, confirm this new connection.
 - 3) Go to 2) until all the GWs (except the m th GW) are examined.
 - 4) If all the hosts in the bottleneck tree are examined, terminate this step. Otherwise, go to 1).
-

In the fifth step, the maximal end-to-end delay is further reduced by exploiting the *frame aggregation*. Specifically, the IEEE 802.11ac frame aggregation assemble multiple frames into one big frame with a single preamble and header information to reduce the transmission overhead. The IEEE 802.11ac introduces Aggregated MAC Protocol Data Unit (A-MPDU) and Aggregated MAC Service Data Unit (A-MSDU) to permit the payload up to 1 Mbyte, which is 16 times larger than its ancestor (IEEE 802.11n).

In this work, the selective routing is permitted to use the frame aggregation to adjust the link speeds so that the AP aggregation can be proposed. Specifically, if multiple hosts are connected to the n th AP, they share the same link between that AP and the associated m th GW. The AP can therefore aggregates the frames from various hosts, and the number of aggregated frames is proportional to the number of hosts using this link. The resulting *effective link speed* \tilde{s}_{nm} after such aggregation is given by

$$\tilde{s}_{nm} = s_{nm} \cdot (1 + \kappa \cdot (|\mathcal{H}_{AP(n)}| - 1)), \quad (9)$$

where, we denote $\mathcal{H}_{AP(n)}$ as the set of hosts connected to the n th AP, and $|\cdot|$ is the cardinality of a set. Therefore, $|\mathcal{H}_{AP(n)}|$ is the number of hosts using the link between n th AP and its connected m th GW. κ represents the coefficient for weighting the effect of aggregated hosts. This effective link speed is then used for the calculation of the end-to-end delay d_m in (4). Intuitively, to reduce the end-to-end delay by increasing the effective link speed with the aid of the frame aggregation, this cross-layer design will increase the number of hosts connecting to the same AP, and eventually reduces the number of active AP as well.

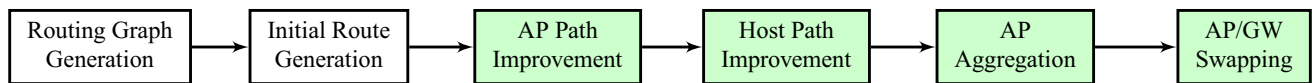


Fig. 4 Routing procedure of the min–max-delay routing

Step 5 (AP Aggregation)

Construct the list of hosts with ascending order according to the estimated host throughput $\hat{s}_{H,k}$ defined in (7).

- ① Select the host with minimum estimated host throughput in the list.
- ② Connect this host with another AP and compute a new maximum end-to-end delay \hat{d}_m by using the effective link speed \hat{s}_{nm} . If $\hat{d}_m < d_m$, confirm this new connection.
- ③ Go to ② until all the APs are examined.
- ④ If all the hosts are examined, terminate this step. Otherwise, go to ①.

Remark 1 Note that since the number of hops between a GW and a host is limited to two, the frame aggregation technique is only applicable to the link between the GW and AP. When the hop limitation of IEEE 802.11 ac commercial devices is removed in the future, the multi-hop relay allows more extensive frame aggregations, leading to much lower maximal end-to-end delay.

In previous steps, the route $\mathcal{R}(\hat{\mathbf{a}})$ is iteratively improved by changing the connection of one host/AP at a time. In this step, the 2-opt algorithm is used to swap the connections between two pairs of APs and hosts for further minimization of the maximal end-to-end delay:

Step 6 (AP/GW Swapping)

- ① Select a pair of hosts using different APs/GWs.
- ② Swap the connected APs/GWs of these two hosts and update the maximum end-to-end delay \hat{d}_m by computing the *effective link speed*. If $\hat{d}_m < d_m$, confirm this swap.
- ③ If all the pair of links between the hosts and different APs/GWs are examined, terminate this step. Otherwise, go to 1).

In summary, the first two steps of the proposed algorithm in Fig. 4 check the feasibility of routing with active APs indicating by $\hat{\mathbf{a}}$. Once a feasible routing can be established, the following fourth steps are applied to reduce the maximal end-to-end delay in $\mathcal{R}(\hat{\mathbf{a}})$. It should be emphasized that, although the AP aggregation seems to reduce the delay by increasing the link speed, this module actually reduces the number of active APs. This is because, more hosts are naturally connected to one AP and consequently, feasible route can be generated with less active APs. Last, for the last four steps, their orders can be

swapped according to various conditions and scenarios to archive better performance. The recursions among these steps may lead to smaller maximal end-to-end delay at the expense of larger algorithmic complexity. Both the step ordering and the recursions are worth for further investigations.

5 WIMNET simulator extension

In this section, we introduce the WIMNET simulator [8] and implement the cross-layer frame aggregation technique in this simulation platform. First, we consider the communication time and the link speed of the IEEE 802.11ac protocol with the frame aggregation. When one 2360 byte data frame is transmitted, the following timing parameters are adopted [23]:

- ① Average waiting time after the end of an interfered link communication: 101.5μ secs
- ② Preamble length: 52μ secs
- ③ Data transmission time: 10.9μ secs
- ④ ACK frame response time : 19.6μ secs

Thus, the accumulated transmission overhead is 173.1μ secs, which includes the first, second, and fourth items enumerated above. When n frames are aggregated into one big frame, the total transmission time of this “big” frame transmission $t(n)$ is given by:

$$t(n) = 173.1 + 10.9n. \quad (10)$$

On the other hand, given the transmitted data size of such big frame

$$r(n) = 2360 \cdot n \quad (11)$$

bytes and the maximum data size for one big frame being 1, 048, 575 bytes, we can compute the maximum number of aggregated frames in one big frame by

$$n_{\max} = \lfloor \frac{1,048,575}{2360} \rfloor = 444, \quad (12)$$

where $\lfloor \cdot \rfloor$ denotes the floor operation. Since the link speed $s(n)$ (Mbps) for this n -frame aggregation is given by

$$s(n) = \frac{r(n)}{t(n)} = \frac{2360n \cdot 8}{173.1 + 10.9n}, \quad (13)$$

the maximum speed of the frame aggregation is thus 1672 Mbps, roughly 96% of the theoretical maximum link speed 1732 Mbps for IEEE 802.11ac.

5.1 Estimations of aggregated frame number of commercial devices

Our WIMNET simulator focuses on practical scenario, where both the maximum link speed L_{\max} and the real-time link speed $s < L_{\max}$ of the commercial devices are changed depending on the environment as shown in Sect. 2.2. The maximum link speed of the devices is generally less than 1732 Mbps due to the smaller number of antennas for the MIMO channel and adopted channel bandwidth. The implementation of the practical WIMNET simulator thus requires the estimation of the number of aggregated frames of a specific commercial device with arbitrary link speed.

The total transmission time of a commercial device to send a big frame aggregating n data frames is inversely proportional to the maximum link speed L_{\max} , i.e., $t(n) \cdot 1732/L_{\max}$. By using this time information and big frame size $r(n)$ in (11), the link speed s of the IEEE 802.11ac devices can alternatively be given by:

$$s = \frac{2360n \cdot 8}{(173.1 + 10.9n) \cdot \frac{1732}{L_{\max}}}. \quad (14)$$

Thus, given the maximum link speed L_{\max} and the actual link speed s of a certain link by using certain device, its number of aggregated frames n can be estimated by:

$$n = \frac{173.1 \cdot 1732 \cdot s}{2360 \cdot 8L_{\max} - 10.9 \cdot 1732 \cdot s} \quad (15)$$

5.2 Number of time slots for one frame transmission

In the WIMNET simulator, all the node operations are synchronized by a single clock called *time slot*. To integrate the frame aggregation in the WIMNET, we not only need to compute the number of aggregated frames for different link speeds, we also need to calculate the number of time slots to transmit one big frame.

As mentioned previously, the time slot for the transmission of a frame without using frame aggregation is 184μ secs ideally when the link speed is 1732 Mbps. With the frame aggregation and using the commercial device, the big frame with n frame aggregated consumes the transmission time $t(n)$ calculated in (10). Normalized by the actual link speed s of the device, the number of required time slots $m(n)$ for the transmission of a big frame is computed as follows:

$$m(n) = \left\lceil \frac{173.1 + 10.9n}{184} \cdot \frac{1732}{s} \right\rceil, \quad (16)$$

where $\lceil \cdot \rceil$ is the ceiling operation. By using this adjusted number of time slots, all the node operations in the WIMNET simulator are synchronized.

6 Simulation results

In this section, we evaluate the proposed selective routing algorithm in different environments by using the WIMNET simulator introduced in Sect. 5 on the platform of Intel (R) Core i7-3770 (3.40 GHz) with Ubuntu 14.04 as virtual OS. Both the uplink and downlink transmissions are simulated. For the uplink, every host randomly transmits 1.25 or 1.5 kbyte packets to the GW, while for the downlink, the GWs sends 10^4 packets to every host. We consider two topologies to respectively represent the open space and indoor environments. To calculate the link speed between two nodes using (8), we assume the information of the node locations is available so that the distance between any pair of nodes can be computed. In practice, the transmission distance and link speed can be estimated by measuring the signal reception strength as well.

First, we consider the indoor topology, as depicted in Fig. 5. This topology models the 2nd floor of engineering building-2 in Okayama University, Japan. There are six rooms with two room sizes: $7\text{m} \times 6\text{m}$ and $3.5\text{m} \times 6\text{m}$. We allocated 60 hosts, 12 APs, and 6 GWs. Each room is equipped with one GW, and (1, 3, 3, 5) APs are distributed to the four larger rooms. The hosts here are regularly distributed since students have their fixed positions in the laboratories. As the wireless signals are degraded when penetrating the concrete wall, we model such phenomenon by reducing the link speed 30% if a wall exists between two nodes [24]. For the AP aggregation, $\kappa = 0.3$ is used to compute the effective link speed defined in (9). For fairness criterion, we set the minimum host throughput threshold $s_T = 50$ Mbps. The routing result is shown in Fig. 5, where the colored markers indicate the active nodes, while the grey markers indicate the inactive ones. We can see that, the proposed selective routing algorithm establish the services for all hosts by using only 4 APs and 6 GWs, only around 56% of the total number of AP/GW.

Next, the numerical experiment results for open-space topology is depicted in Fig. 6, where 50 hosts, 20 APs, and 6 GWs are distributed in a $100\text{m} \times 120\text{m}$ rectangular area. Since the GWs are connected to the Internet through wired Ethernet, they are placed in the corners. The APs are equally distributed in the grids of this area. We set $(\kappa, s_T) = (0.3, 50)$ as in the indoor topology. From Fig. 6, we see that only 3 APs and 2 GWs are activated, around 19% of the total APs/GWs. The percentage of active APs/GWs in the open-space topology is much smaller than that in the indoor topology, because the environment of the later is split into multiple isolated rooms, and the hosts are clustered around APs and GWs, while the hosts are more uniformly distributed and no walls

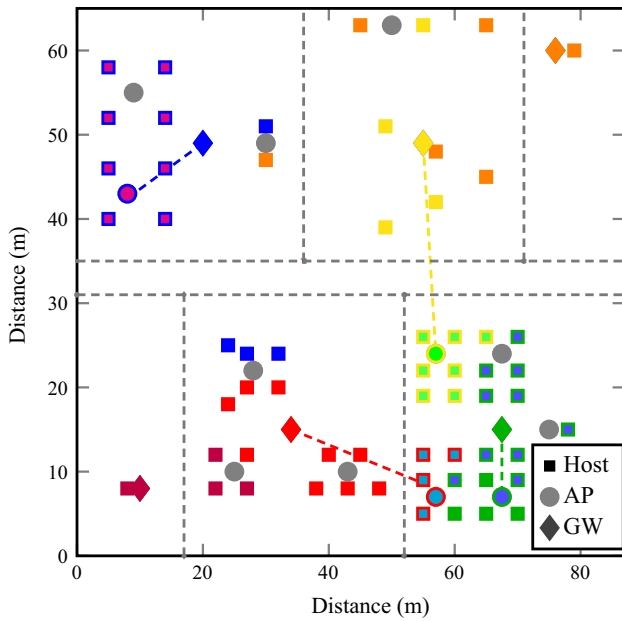


Fig. 5 The simulation results of the proposed selective routing algorithm in the indoor topology with $(\kappa, sT) = (0.3, 50)$. *Square, circle, and diamond* are respectively used to represent the host, AP, and GW. For each marker, *edge color* indicates the connection to the GW with the same color, while the *fill color* indicate the connection to the AP or GW with the same color. Those *grey* markers are inactive APs/GWs. *Dashed lines* represent the connections between the AP/GW

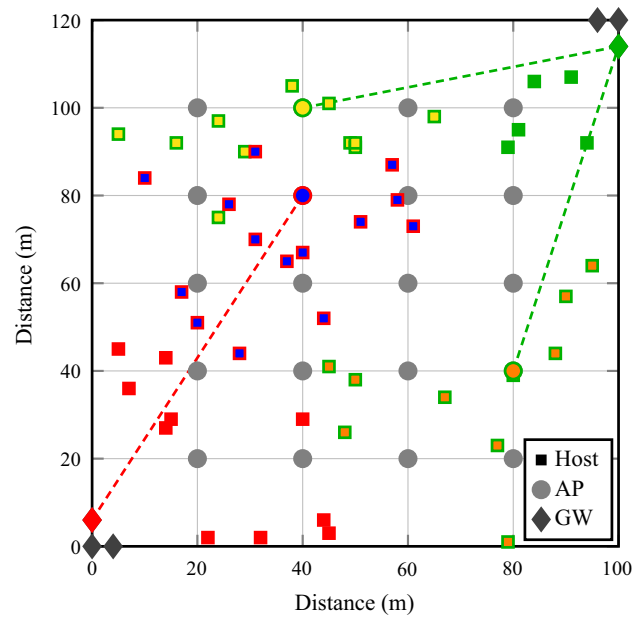


Fig. 6 The simulation results of the proposed selective routing algorithm in the open-space topology with $(\kappa, sT) = (0.3, 50)$. *Square, circle, and diamond* are respectively used to represent the host, AP, and GW. For each marker, *edge color* indicates the connection to the GW with the same color, while the *fill color* indicate the direct connection to the AP or GW with the same color. Those *grey* markers are inactive APs/GWs. *Dashed lines* represent the connections between the AP/GW

appear in the former environments. The numerical experiment results in Figs. 5 and 6 implies that, although selective routing algorithm works well in both environments, more APs/GWs can be deactivated in places like public squares, school campus, or metro stations. This is a big advantage, since in these places the number of AP/GWs are generally numerous and hence the selective routing can greatly reduce the operational costs of the authority.

Now, to demonstrate the effectiveness of the selective routing algorithm, Fig. 7 depicts the number of active APs/GWs with respect to the proposed selective routing with various values of κ . We also consider a *simple selective routing* where all the steps that enhance the routing performance are removed. Specifically, we removed the *plural AP deactivation, AP path improvement, host path improvement, AP aggregation, AP/GW swapping, and AP/GW deactivation*. With $\kappa = 0.3$, the proposed algorithm can deactivate additional 17 and 5 devices in the open-space topology and the indoor topology, respectively. Such improvement demonstrates the effectiveness of the proposed algorithm. For the cross-layer AP aggregation, the selective routing algorithm can establish the route with less active APs /GWs for larger κ , since the effective link speeds increases accordingly. Compared with the cases of no AP aggregation ($\kappa = 0.2$), 3 and 2 more devices are

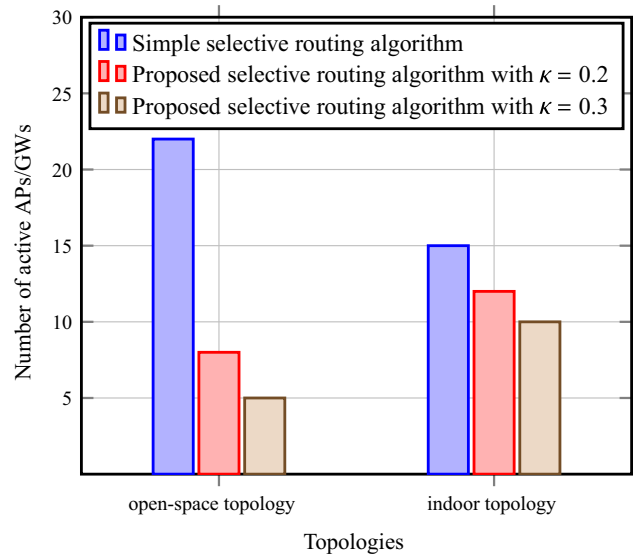


Fig. 7 The effectiveness of the proposed selective routing algorithm for the indoor and open-space topologies

deactivated when $\kappa = 0.3$ in the open-space topology and indoor topology, respectively. After this saturation, increasing frame aggregation rate does not have any effect on the performance.

To examine if the maximum end-to-end delay has been increased due to less active APs/GWs, Fig. 8 depicts the results during the iterations of the selective routing. For clarity, the plural AP deactivation is turned-off so that we can see the trends of the delay as the number of AP gradually decreases. For the indoor topology, the proposed selective routing keeps the delay slightly fluctuates in all cases. For the open-space topology, the delay is nearly the same when the number of active APs/GWs is larger than 20, and increases around 25% when the number of active APs/GWs is 10. When all 50 hosts are supported by only 5 active APs/GWs, the maximum delay increases significantly. This is because, as in practice an AP/GW supports a finite number of hosts, we set this *maximal number of associated hosts* to 14 in our simulator. When an AP/GW reaches this value, its transmission latency increases 50% to model the congestion and nonlinear queueing delay, which is the case of the AP located on (40, 80) and two active GWs in Fig. 6.

Nonetheless, even in such extreme case, the maximum delay is less than twice of the case without any deactivations, demonstrating that the AP/GW are nicely selected to reduce the active devices and delays as well. The reason for the fluctuations observed in Fig. 8 is that, the proposed min-max delay routing in Sect. 4 reduces the delay by heuristically changing the local connections between the hosts and APs/GWs. Therefore, the global optimality is not guaranteed. When the AP associated with the maximal end-to-end delay is deactivated, since the hosts are re-distributed to other APs/GWs, the new routing may result in lower maximal end-to-end delay.

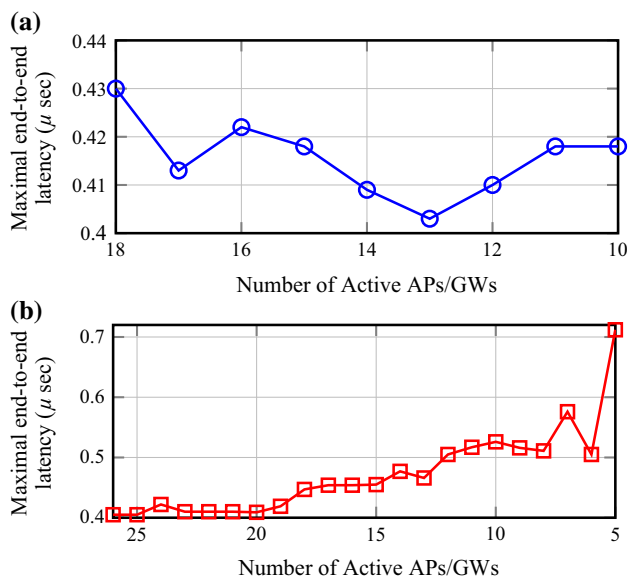


Fig. 8 Number of active APs /GWs versus maximum latency for indoor and open-space topologies with $\kappa = 0.3$

Table 2 Comparison of the algorithm run time with minimum host throughput and plural AP deactivation preprocessing in both open-space topology and indoor topology

CPU time (sec)		
Plural AP deactivation	×	○
Indoor topology	49.46	27.20
Open-space topology	55.65	41.53

Last, the performance of the plural AP deactivation is demonstrated in Table 2 by measuring the CPU simulation time used to achieve the routing results in Figs. 5 and 6. We can see that the plural AP deactivation drastically reduces the run time. When applying the plural AP deactivation, the routing is generated by using around 55 and 74.6% CPU time in indoor and open-space topologies, respectively. The reason that the plural AP deactivation saves more time in the indoor topology is because the APs in the larger rooms are close to each other, and hence can be deactivated simultaneously by using the plural AP deactivation. It should be emphasized that, although the effectiveness of the proposed algorithm is demonstrated by the numerical experiments, the theoretical analysis that sheds light on the routing design is still worth of further investigation in the future.

7 Conclusion

In this work, we propose the selective routing algorithm for the joint minimization of the number of active APs/GWs (operational cost) and the end-to-end transmission delay. This algorithm is investigated for IEEE 802.11ac WIMNET with six practical constraints, e.g., fairness criterion. A hierarchical optimization is formulated, and the algorithm comprising preprocessing module, iterative module, and postprocessing module, is designed. The preprocessing is designed to speed up the algorithm, which saves around half CPU time. In the main iterative module, the APs and GWs are sequentially deactivated over iterations. In each iteration, the Dijkstra algorithm is applied to check the routing feasibility. The local searches like 2-opt algorithm are applied to refine the routing in the sub-modules of min-max-delay routing module, and to further reduce the number of active APs in the post-processing module, respectively. Together with the cross-layer AP aggregation, the proposed algorithm only activates 19% of the APs/GWs to accomplish the routing in open space environment. We believe that the proposed selective routing opens a new door for the routing design, which especially benefits the

large-scale networks where numerous nodes causes huge operational costs. In this case, the proper design of the AP/GW deactivations becomes more important. Moreover, while WIMNET is considered in this network, different network architectures require the re-innovations of the selective routing. For example, in the fiber-wireless (FiWi) network which integrates passive optical network and wireless network [25, 26], we have heterogeneous devices that can be activated/deactivated. Also, the multihop wireless relay path can generally be replaced by wireless-optical-wireless path. In this case, a more sophisticated selective routing algorithm should be designed. The wireless network with high mobility is another interesting scenario, since the APs may be more frequently activated and deactivated. In the future, we will consider other performance metrics, e.g., jitter and average/minimum user throughput for the online multimedia streaming services. We also propose to increase the scope of the cross-layer design, which means that other technologies, e.g., MIMO and MU-MIMO, will be incorporated and jointly designed in the routing algorithm, realizing a layerless routing design.

Acknowledgements The work of I-Wei Lai was supported by Ministry of Science and Technology (MOST) under the grant MOST 105-2221-E-182-006.

References

- Crow, B. P., Widjaja, I., Kim, J. G., & Sakai, P. T. (1997). IEEE 802.11 wireless local area networks. *IEEE Communications Magazine*, 35(9), 116–126.
- Gast, M. S. (2005). *802.11 wireless networks—the definitive guide* (2nd ed.). Sebastopol: O’Reilly.
- Funabiki, N. (2011). Wireless mesh networks. InTech-Open Access Publisher. <http://www.intechopen.com/books/show/title/wireless-mesh-networks>.
- Raniwala, A., & Chiueh, T.-C. (2005). Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *Proceedings of IEEE INFOCOM*, (pp. 2223–2234).
- Funabiki, N., Maruyama, W., Nakanishi, T. & Watanabe, K. (2012). An extension of routing tree algorithm considering link speed change in IEEE 802.11n protocol for wireless mesh network. In *Proceedings of MENS2012*, (pp. 600–605).
- Hassan, W., Funabiki, N., & Nakanishi, T. (2010). Extensions of the access point allocation algorithm for wireless mesh networks. *IEICE Transactions on Communications*, E93–B(6), 1555–1565.
- Zhang, Y., Luo, J., & Hu, H. (Eds.). (2007). *Wireless mesh networking: Architectures, protocols and standards*. New York: Auerbach.
- Funabiki, N., Shimitzu, J., Nakanishi, T., & Watanabe, K. (2011). A proposal of an active access-point selection algorithm in wireless mesh networks. In *Proceedings of international conference on network-based information systems (NBIS 2011)*, (pp. 112–117).
- Bergamo, P., Giovanardi, A., Travasoni, A., Maniezzo, D., Mazzini, G., & Zorzi, M. (2004). Distributed power control for energy efficient routing in ad hoc networks. *Wireless Networks*, 10(1), 29–42.
- Mohsenian-Rad, A. H., & Wong, V. W. S. (2007). Joint logical topology design, interface assignment, channel allocation, and routing for multi-channel wireless mesh networks. *IEEE Transactions on Wireless Communications*, 6(12), 4432–4440.
- Sasikala, T., Bhagyaveni, M. A., Senthil, K., & Jawahar, V. (2016). Cross layered adaptive rate optimised error control coding for WSN. *Wireless Networks*, 22(6), 2071–2079.
- Akyildiz, I. F., & Wang, X. (2008). Cross-layer design in wireless mesh networks. *IEEE Transactions on Vehicular Technology*, 57(2), 1061–1076.
- Li, K., & Wang, X. (2008). Cross-layer design of wireless mesh networks with network coding. *IEEE Transactions on Mobile Computing*, 7(11), 1363–1373.
- Benyamina, D., Hafid, A., & Gendreau, M. (2012). Wireless mesh networks design—A survey. *IEEE Communications Surveys and Tutorials*, 14(2), 299–310.
- Karkazis, P., Trakadas, P., Leligou, H. C., Sarakis, L., Papaefstathiou, I., & Zahariadis, T. (2013). Evaluating routing metric composition approaches for QoS differentiation in low power and lossy networks. *Wireless Networks*, 19(6), 1269–1284.
- Yang, X., & Chen, B. (2016). A novel method for measurement points selection in access points localization. *Wireless Networks*, (pp. 1–14). doi:10.1007/s11276-016-1315-y
- Wu, Y., Zhang, Z., Wu, C., Li, Z., & Lau, F. (2013). CloudMoV: Cloud-based mobile social TV. *IEEE Transactions on Multimedia*, 15(4), 821–832.
- Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Norwell: Kluwer.
- Lai, Y.-J. (1996). Hierarchical optimization: A satisfactory solution. *Fuzzy Sets and Systems*, 77(3), 321–335.
- Chiueh, T.-D., Tsai, P.-Y., & Lai, I.-W. (2012). *Baseband receiver design for wireless MIMO-OFDM communications* (2nd ed.). New York: Wiley-IEEE Press.
- Chew, C. C., Funabiki, N., Maruyama, W., & Fujita, S. (2014). An extended active access-point selection algorithm for link speed changes in wireless mesh networks. *International Journal of Space-Based and Situated Computing*, 4(3–4), 184–193.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271.
- Asai, Y., et al. (2014). Overview of very high throughput wireless LAN standard IEEE 802.11ac and experimental evaluation of multiuser-MIMO transmission. *IEICE Transactions on Communications*, J97–B(1), 1–18.
- Takebayashi, T. (2012). Throughput measurements of IEEE 802.11n wireless LAN and application to network design. Masters thesis, Graduate School of Natural Science and Technology, Okayama University, Japan.
- Zheng, Z., & Wang, J. (2009). A study of network throughput gain in optical-wireless (FiWi) networks subject to peer-to-peer communications. In *International conference on communications*, (pp. 1–6).
- Aurzada, F., Lvesque, M., Maier, M., & Reisslein, M. (2014). FiWi access networks based on next-generation PON and gigabit-class WLAN technologies: a capacity and delay analysis. *IEEE/ACM Transactions on Networking (ToN)*, 22(4), 1176–1189.



I-Wei Lai received his Ph.D. in electrical engineering from the Graduate Institute of Electronics Engineering (GIEE) at National Taiwan University in 2011, and he is now an Assistant Professor in Department of Electrical Engineering, National Taiwan Normal University (NTNU), Taiwan. He was a Research assistant with the Institute of Integrated Signal Processing Systems, RWTH Aachen University, Germany. He was the recipient of the NTU-GIEE

best thesis award in 2011 for his Ph.D. thesis and is a member of Phi Tau Phi Scholastic Honor Society. His research interests include baseband signal processing, optimization, MIMO communication, and ad hoc cognitive radio network. He was recognized as an Exemplar Reviewer of IEEE Transactions on Communications, 2014.



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 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, and Web technology. 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. degrees in information science and technology from the 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 a research associate in 1996. His

research interests include computer network and optimization algorithm.



Md. Selim Al Mamun received his B.Sc. (Hons) and M.S. degrees in Computer Science and Engineering from the University of Dhaka, Bangladesh in 2007 and 2009 respectively. He joined as a faculty member at the Department of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh in 2011. He is currently pursuing the Ph.D. degree at the Graduate School of Natural Science and Technology,

Okayama University, Japan. His research interests include computer networks, optimization algorithms, network security and quantum computing.



Sho Fujita received his B.S. and M.S. from the Department of Electrical and Communication Engineering, Okayama University in 2013 and 2015 respectively. His research interests include computer networks and artificial intelligence.