Received 1 December 2014; revised 13 February 2015; accepted 5 May 2015.
Date of Publication 14 May 2015; date of current version 8 June 2016.

Digital Object Identifier 10.1109/TETC.2015.2433936

Efficient Multicast Delivery for Data Redundancy Minimization Over Wireless Data Centers

CHING-CHIH CHUANG[1], (Student Member, IEEE), YA-JU YU[2], AI-CHUN PANG[1,3,4], (Senior Member, IEEE), HSUEH-WEN TSENG[5], (Member, IEEE), and HSIN-PENG LIN[1,6]

[1]Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan
[2]Smart Network System Institute, Institute for Information Industry, Taipei 106, Taiwan
[3]Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan
[4]Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 10617, Taiwan
[5]Department of Computer Science and Engineering, National Chung Hsing University, Taichung 402, Taiwan
[6]Telecommunication Laboratories, Chunghwa Telecom Company, Ltd., Taipei 235, Taiwan

CORRESPONDING AUTHOR: A.-C. PANG (acpang@csie.ntu.edu.tw)

This work was supported in part by the Excellent Research Projects of National Taiwan University under Grant 104R890822, in part by the Ministry of Science and Technology under Grant 102-2221-E-002-075-MY2, Grant 103-2221-E-002-142-MY3, and Grant 102-2221-E-005-037-MY2, in part by the Information and Communications Research Laboratories, in part by the Industrial Technology Research Institute, in part by the Institute for Information Industry, and in part by the Research Center for Information Technology Innovation, Academia Sinica.

ABSTRACT With the explosive growth of cloud-based services, large-scale data centers are widely built for housing critical computing resources to gain significant economic benefits. In data centers, the cloud services are generally accomplished by multicast-based group communications. Recently, many well-known industries, such as Microsoft, Google, and IBM, adopt high-speed wireless technologies to augment network capacity in data centers. However, those well-known multicast delivery schemes for traditional wired data centers do not consider the unique characteristics of wireless communications, which may result in unnecessary data transmissions and network congestions. Under the coexisting scenario of wired and wireless links, this paper studies multicast tree construction and maintenance problems. The objective is to minimize the total multicast traffic. We prove the problems are $\mathcal{NP}$-hard and propose efficient heuristic algorithms for the two problems. Based on real traces and practical settings obtained from commercial data centers, a series of experiments are conducted, and the experimental results show that our proposed algorithms are effective for reducing multicast data traffic. The results also provide useful insights into the design of multicast tree construction and maintenance for wireless data center networks.

INDEX TERMS Data redundancy, multicast, wireless data centers.

I. INTRODUCTION

With the explosive growth of cloud-based services, large-scale data centers are widely built for housing critical computing resources to gain significant economic benefits. In data center networks, the cloud-based services are mostly accomplished by group communications with multicast traffic. For instance, a web server redirects queries to a set of indexing servers. Distributed file systems replicate file chunks to a set of storage nodes [1]. For distributed execution engines such as MapReduce [2], the master node assigns tasks to a group of servers for cooperative computations. In social networks (e.g., Facebook, Twitter, etc) [3], users frequently share their messages, photos and videos with their friends, and group communications are also needed. In group communications, a source node has to transmit one copy of the data to multiple destination nodes. If the same data is dispersedly transmitted by different links to different destinations, the multicast traffic will occupy a large portion of network resources, which results in network congestions. According to the measurements reported by Microsoft, the number of multicast groups in a data center is large and each group generally comprises numerous multicast members [4]; the data traffic in top-of-rack switches is heavy and may cause serious degradation in network performance [5].

To effectively accommodate the huge amount of data traffic in data center networks, high-speed wireless technologies (e.g., 802.11ad 60GHz wireless transmissions) are considered, in existing wired data centers such as Microsoft [6], Google [7], and IBM [8], to be used on top-of-rack switches to augment network capacity and provide fast connectivity. Specifically, in [9], a comprehensive analysis demonstrates that the hybrid structure, where wireless access points and wired switches coexist, is a feasible solution for data centers. In such the wireless data center, multicast data can be transmitted by either wireless access points or wired switches. Although wireless medium is broadcast in nature and might be more suitable for multicast, how to build multicast trees in wireless data centers is complicated and faces many challenges. The challenges mainly come from the following factors. 1) Since wireless access points are densely deployed in data centers, the interference issue among wireless access points should be carefully considered. 2) Unlike a wired switch, a wireless access point can transmit data to more than one access point in its communication range and has more selections for transmission paths, especially when a directional antenna is adopted [5]. 3) The coexistence of wired and wireless links lead to the interesting issue that how to avoid wireless interference by adopting wired links in wireless data centers such that more wireless access points can be transmitted simultaneously.

In addition to the above challenges, the cloud services such as social networks and VM migration have some receivers dynamically joining and leaving their multicast groups so their multicast trees have to be reconstructed when the events occur. The tree reconstruction in this case will cause a "chain reaction". That is, the changes will be made not only for the groups (abbreviated as "involved groups") with member joining and leaving, but also for the groups (abbreviated as "victim groups") which are affected by "involved groups" due to wireless interference. A trivial way to avoid wireless interference is to switch the affected transmissions from wireless to wired links, which will definitely generate a large amount of redundant multicast data traffic. Alternatively, an exhausting computation and excessive signaling exchanges for overall tree reconstruction need to be done to minimize the redundancy. Thus how to efficiently transmit multicast data while maintaining low computation without involving too many multicast trees should be carefully studied. We will give two simple examples in Section III to respectively describe the above mentioned challenging issues for wireless data center networks in more details.

In this paper, we address the group communication issues, multicast tree building and maintenance, raised in wireless data center networks comprised of wired and wireless links. The objective is to minimize the total multicast data traffic. The contributions of this paper are as follows. Firstly, we formulate the multicast tree building and maintenance problems with the consideration of coexisting wired and wireless links in wireless data center networks. We prove that the target problems are $\mathcal{NP}$-hard. For the tree building problem, we propose a heuristic algorithm to efficiently use wireless transmission links. For the tree maintenance problem, a low-complexity solution is presented to reconstruct the multicast trees when receivers join or leave. Finally, we conduct a series of simulations based on practical parameter settings to evaluate the performance of our proposed algorithms. We collect real traces of MapReduce from the largest telecom operator in Taiwan and refer to their data center topology for our simulation setup. The simulation results demonstrate that our proposed algorithms are very effective in reducing the total data redundancy of the multicast traffic. The results also provide useful insights into the design of multicast tree building and maintenance for wireless data center networks.

The rest of the paper is organized as follows. In Section II, we review some related works on multicast tree construction and maintenance. Section III describes our system model and formulates the problems. In Section IV and V, we prove that our target problems are $\mathcal{NP}$-hard and propose efficient heuristic solutions. Simulation results are presented in Section VI. Section VII concludes the paper.

II. RELATED WORKS

To achieve group communications, multicast is used to transmit data to a group of destinations. The first standard of IP (Internet Protocol) multicast is specified in RFC 1112 [10]. Then the Internet Group Management Protocol (IGMP) is defined to allow a host to join and leave a group, and to report its IP multicast group membership to neighboring multicast routers [11]. The tree structure is commonly adopted for multicast to reduce redundant data transmissions and avoid unnecessary network resource usage. The multicast tree can be built by the two methods, source-based and share-based [12]. The source-based tree is established by the shortest-path algorithm, and each sender requires an individual tree to transmit its multicast data. This implies that the source-based multicast tree is more suitable for the applications with few senders in a multicast group. In contrast, only one shared-based tree is needed for a multicast group. Multiple senders in a common multicast group can share the tree. However, for both source-based and shared-based multicast trees, the tree establishment and maintenance procedures generally follow the receiver-driven manner, which would result in redundant transmission links especially when there are multiple disjoint equal-cost paths between a pair of servers in wired data center networks [13].

For wireless ad-hoc networks, multicast routing has been widely studied [14], and can be roughly classified into tree-based, mesh-based, and hybrid-based approaches. The tree-based approach establishes a single path between any two nodes in a multicast group [15]. Since ad-hoc nodes can move freely, the tree needs to be frequently re-established due to link failure such that packet delivery ratio is decreased. Thus, some studies, see [16], proposed the meshed-based approach to provide multiple paths for robust connectivity for group communications. However, massive control messages used to update topology information and redundant paths consume a large portion of network resources. Consequently, hybrid-based multicast routing protocols, see [17], were proposed. The above wireless multicast routing approaches cannot be applied to wireless data center networks, since they do not consider how to build and maintain multicast trees when wired and wireless links co-exist.

Recently, some researches have paid attention to multicast issues in traditional wired data centers. In [18], considering the hardware constraint in supporting multicast operations in switches, Vigfusson et al. developed a mechanism to select parts of group communication requests to adopt multicast delivery while the remaining requests are accomplished by unicast transmissions. Then, Li et al. [13], [19] observed that the receiver-driven multicast routing protocols designed for the Internet do not perform well in terms of the number of transmission links in densely connected data center networks with multiple disjoint equal-cost paths. Thus, to reduce data transmission redundancy for wired data center networks, an efficient multicast tree establishment and maintenance approaches were presented for the case that receivers can dynamically join or leave a multicast group. However, the approaches do not take wireless links into account, and only reduce the total number of used wired links, as their major performance metric, without considering different data rates requested by heterogeneous cloud services.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM MODEL

In a data center, several servers are grouped in a rack and each rack is equipped with a switch. The switch is named as the top-of-rack switch which connects to all the servers in the rack. Top-of-rack switches are generally connected by aggregation switches and/or core switches, depending on their network topology. The types of data center network topology include hierarchical topology, Fat-tree [20] and BCube [21]. Considering the deployment cost and complexity of wired links, hierarchical topology is commonly used. Moreover, many industries [5], [7], [8] are trying to deploy access points with 60GHz wireless access technologies on top-of-rack switches to augment network capacity and provide fast connectivity. The 60GHz access points can support high data rate with the transmission range of 10 meters. Since the density of access points is extremely high in data centers, the access points are generally equipped with the directional narrow-beam antenna array to mitigate interference [6]. Under a managed environment, we assume that a data center will have a central controller to manage the forwarding table of switches. The illustration of a simple *wireless data center* architecture is shown in Fig. 1, where there are twelve racks, and each rack has one top-of-rack switch and one wireless access point. Each top-of-rack switch connects to an aggregation/core switch by the wired link, while each top-of-rack access point can transmit data to any access point within its transmission range.

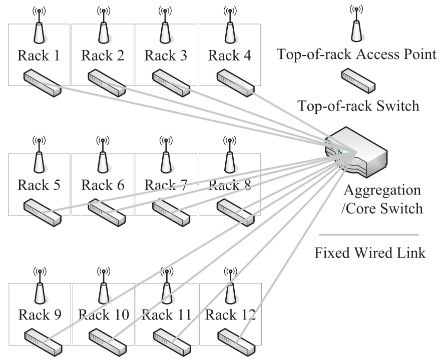


FIGURE 1. A simple wireless data center architecture.

In wireless data centers, multicast data traffic is delivered frequently, and tree-based transmission is an effective way to accomplish the multicast delivery. However, how to build and maintain multicast trees under the co-existence of wired and wireless links to minimize redundant multicast traffic in wireless data centers is still open and challenging. When multicast groups are created, we have to construct the corresponding multicast trees for the groups, referred to as *multicast tree construction problem*. On the other hand, when receivers join or leave a multicast group which has already existed, we have to reconstruct/maintain the multicast tree, referred to as *multicast tree maintenance problem*. The approaches for constructing and maintaining multicast trees can be classified into two types [12], source-based and share-based. Since most of the group communications in data centers have only one multicast sender, without loss of generality, this paper adopts the source-based approach.

B. PROBLEM FORMULATION

In this paper, we are interested in the source-based multicast tree construction and maintenance, comprised of wired and wireless links in data center networks. The objective is to minimize the total multicast data traffic (i.e., the transmission redundancy). The problem formulation is described as follows. For the sake of brevity, we omit "∀" when the meaning is clear from the context.

1) THE MULTICAST TREE CONSTRUCTION PROBLEM

A wireless data center is modeled as a directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$. The $\mathbb{V} = (\mathbb{V}^F, \mathbb{V}^W)$ is a set of racks. Each rack $v \in \mathbb{V}$ includes one top-of-rack switch $s_v \in \mathbb{V}^F$ and one wireless access point $a_v \in \mathbb{V}^W$. The $\mathbb{V}^F$ is a set of top-of-rack switches and $\mathbb{V}^W$ is a set of top-of-rack access points. The link set $\mathbb{E} = (\mathbb{E}^F, \mathbb{E}^W)$ includes a set of wired (fixed) links $\mathbb{E}^F$ and a set of wireless links $\mathbb{E}^W$. Wired link $e^F_{s_i s_j} \in \mathbb{E}^F$ with capacity $C^F_{s_i s_j}$ (bps) represents that top-of-rack switch s_i can transmit data to top-of-rack switch s_j by the wired link. On the other hand, wireless link $e^W_{a_x a_y} \in \mathbb{E}^W$ with capacity $C^W_{a_x a_y}$ (bps) indicates that access point a_x can transmit data to access point a_y by the wireless link.

We consider a set of N multicast groups $\mathbb{R}=(r_1, r_2, \ldots, r_N)$, where $r_k = (v_k, \mathbb{D}_k, T_k)$ means that rack v_k is the sender of multicast group k and has to transmit the multicast traffic with data rate T_k (bps) to a set of destinations (racks) $\mathbb{D}_k \subseteq \mathbb{V}$. Then, we define $l^F(k, e^F_{s_is_j}) \in \{0, 1\}$ as an indicator function, which registers 1 if the traffic of multicast group k passes through wired link $e^F_{s_is_j}$. If wired link $e^F_{s_is_j}$ is used and $l^F(k, e^F_{s_is_j})$ is set at 1, top-of-rack switch s_j of rack $j \in \mathbb{V}$ can receive the multicast data of group k. We also define $l^W(k, e^W_{a_xa_y}) \in \{0, 1\}$ to indicate whether the traffic of multicast group k uses wireless link $e^W_{a_xa_y}$ or not. If wireless link $e^W_{a_xa_y}$ is selected and $l^W(k, e^W_{a_xa_y})$ is set at 1, the set of access points of racks $\mathbb{S}_{a_xa_y} \subset \mathbb{V}$ within the coverage area of the transmission can overhear and receive the data. Our purpose is to build a multicast tree, comprised of wired and wireless links, for each multicast group.

2) THE MULTICAST TREE MAINTENANCE PROBLEM

After the multicast trees are constructed, the problem is to adjust the tree structure when there are receivers requesting to join or leave their multicast groups. In addition to the inputs of the tree construction problem, the tree maintenance problem are further described as follows. A set of racks J_k and L_k respectively has nodes requesting to join and leave multicast group k. Thus, the set of destinations $\mathbb{D}_k$ of multicast group k is changed to $(\mathbb{D}_k \cup J_k) \setminus L_k$. Given the wired and wireless link indicator functions $l^F(k, e^F_{s_is_j})$ and $l^W(k, e^W_{a_xa_y})$ determined in the multicast tree construction problem, we have to maintain the multicast tree with wired $\hat{l}^F(k, e^F_{s_is_j})$ and wireless link indicator function $\hat{l}^W(k, e^W_{a_xa_y})$ for each new set of destinations $\mathbb{D}_k$.

The solutions for the above multicast tree construction and maintenance are *feasible* if the following constraints are met. Note that $l^F(k, e^F_{s_is_j})$ and $l^W(k, e^W_{a_xa_y})$ in Equations (1)-(3) is respectively replaced by $\hat{l}^F(k, e^F_{s_is_j})$ and $\hat{l}^W(k, e^W_{a_xa_y})$ when the tree maintenance problem is considered.

a: WIRED LINK CAPACITY CONSTRAINT

In order to avoid over-utilization of top-of-rack switches, Equation (1) ensures that the data rate of multicast group through each wired link cannot exceed the available capacity of each wired link.

$$\sum_{k=1}^{N} T_k \cdot \left[l^F(k, e^F_{s_is_j}) + l^F(k, e^F_{s_js_i}) \right] \le C^F_{s_is_j}, \quad \forall e^F_{s_is_j} \in \mathbb{E}^F. \quad (1)$$

b: ACCESS POINT CAPABILITY CONSTRAINT

Since wireless access points incurs interference from their neighboring access points, Equation (2) states that each access point cannot exceed its capability including interference/data reception (first term) and transmission (second term). $I(a_y, e^W_{a_xa_z})$ is used to indicate whether access point a_y is interfered by access point a_x, and defined based on a geometric-based protocol interference model [22]. Based on the protocol interference model, $I(a_y, e^W_{a_xa_z}) = 1$ when access point a_y is located in the transmission range of access point a_x for delivering data to access point a_z.

$$\sum_{k=1}^{N} \sum_{a_x \in \mathbb{V}^W} \sum_{a_z \in \mathbb{V}^W} \Big(\frac{I(a_y, e^W_{a_xa_z}) l^W(k, e^W_{a_xa_z}) T_k}{C^W_{a_xa_z}} + \frac{l^W(k, e^W_{a_ya_x}) T_k}{C^W_{a_ya_x}} \Big) \le 1, \quad \forall a_y \in \mathbb{V}^W, \quad a_x \ne a_z \quad (2)$$

where

$$I(a_y, e^W_{a_xa_z}) = \begin{cases} 1, & \text{if } y \in \mathbb{S}_{a_xa_z} \\ 0, & \text{otherwise.} \end{cases}$$

c: DELIVERY CONSTRAINT

The destinations of each multicast group must receive their multicast data.

$$\left(\bigcup_{l^W(k, e^W_{a_xa_y})=1} \mathbb{S}_{a_xa_y} \right) \bigcup \left(\bigcup_{l^F(k, e^F_{s_is_j})=1} j \right) \supseteq \mathbb{D}_k, \quad \forall r_k \in \mathbb{R} \quad (3)$$

We now define the target problem formally as follows.

3) THE EFFICIENT MULTICAST TREE CONSTRUCTION PROBLEM

Input instance: Consider a directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$. Each wired and wireless link has its capacity $C^F_{s_is_j}$ and $C^W_{a_xa_y}$. There is a set of N multicast groups $\mathbb{R}$.

Objective: Our objective of this problem is to build a multicast tree, comprised of wired $l^F(k, e^F_{s_is_j})$ and wireless links $l^W(k, e^W_{a_xa_y})$, for each multicast group such that the multicast data traffic (data redundancy) of all multicast groups is minimized. The objective function is expressed as follows.

$$\text{Min} \sum_{k=1}^{N} \sum_{e^F_{s_is_j} \in \mathbb{E}^F} \sum_{e^W_{a_xa_y} \in \mathbb{E}^W} T_k \times \left[l^F(k, e^F_{s_is_j}) + l^W(k, e^W_{a_xa_y}) \right],$$

subject to constraints (1)-(3).

4) THE EFFICIENT MULTICAST TREE MAINTENANCE PROBLEM

Input instance: Consider a directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$. Each wired and wireless link has its capacity $C^F_{s_is_j}$ and $C^W_{a_xa_y}$. There is a set of N multicast groups $\mathbb{R}$. Given the tree structure of each multicast group (i.e., wired $l^F(k, e^F_{s_is_j})$ and wireless links $l^W(k, e^W_{a_xa_y})$), each multicast group k has a set of nodes J_k and L_k requesting to join and leave.

Objective: Our objective of this problem is to maintain each multicast tree, comprised of wired $\hat{l}^F(k, e^F_{s_is_j})$ and wireless links $\hat{l}^W(k, e^W_{a_xa_y})$, for the set of joining and leaving nodes such that the increased multicast data traffic of all multicast groups is minimized. The objective function is expressed as follows.

$$\text{Min} \sum_{k=1}^{N} \sum_{e^F_{s_i s_j} \in \mathbb{E}^F} \sum_{e^W_{a_x a_y} \in \mathbb{E}^W} T_k \times \left[\hat{l}^F(k, e^F_{s_i s_j}) + \hat{l}^W(k, e^W_{a_x a_y})\right]$$
$$- \sum_{k=1}^{N} \sum_{e^F_{s_i s_j} \in \mathbb{E}^F} \sum_{e^W_{a_x a_y} \in \mathbb{E}^W} T_k \times \left[l^F(k, e^F_{s_i s_j}) + l^W(k, e^W_{a_x a_y})\right],$$

subject to constraints (1)-(3). Table 1 summarizes the notations used in the problem formulation.

TABLE 1. Summary of notations.

$\mathbb{V}$	The set of racks
$\mathbb{V}^F$	The set of top-of-rack switches
$\mathbb{V}^W$	The set of top-of-rack access points
$\mathbb{E}^F$	The set of wired links
$\mathbb{E}^W$	The set of wireless links
$C^F_{s_i s_j}$	The available capacity of wired link $e^F_{s_i s_j}$
$C^W_{a_x a_y}$	The available capacity of wireless link $e^W_{a_x a_y}$
$\mathbb{R}$	The set of multicast groups
$\mathbb{D}_k$	The set of destinations of multicast group k
T_k	The data rate of multicast group k
$\mathbb{S}_{a_x a_y}$	The set of access points can overhear the multicast data, when access point a_x transmits data to access point a_y
J_k	The set of racks has nodes requesting to join multicast group k
L_k	The set of racks has nodes requesting to leave multicast group k
$l^F(k, e^F_{s_i s_j})$	An indicator function, which is 1 if wired link $e^F_{s_i s_j}$ is allocated to multicast group k, and 0 otherwise
$l^W(k, e^W_{a_x a_y})$	An indicator function, which is 1 if wireless link $e^W_{a_x a_y}$ is allocated to multicast group k, and 0 otherwise

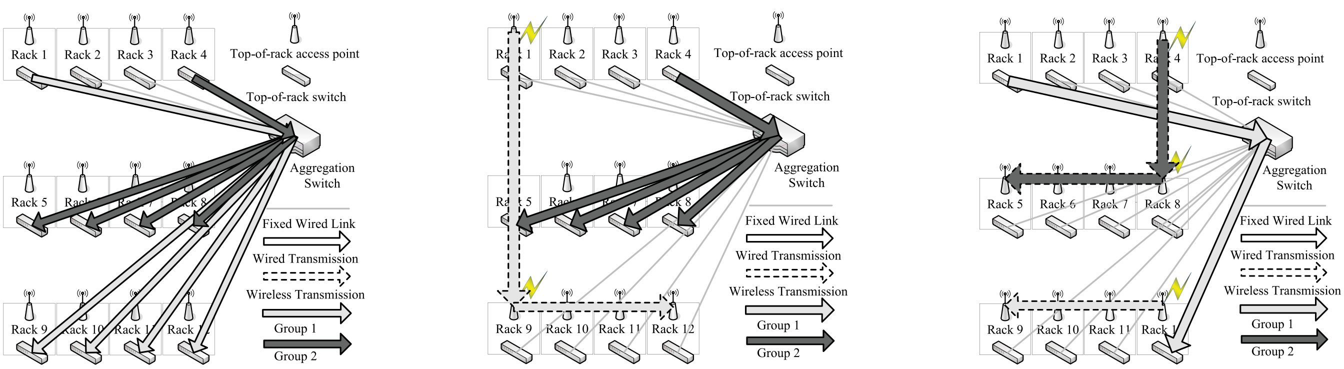
C. AN ILLUSTRATIVE EXAMPLE

1) MULTICAST TREE CONSTRUCTION

We use a simple example, as shown in Fig. 2, to describe the multicast tree construction problem in wireless data centers. Consider the wireless data center $\mathbb{G}$ shown in Fig. 1. On each rack, there is a pair of top-of-rack switch and access point. The data sent from one top-of-rack switch to another should go through two wired links, while a top-of-rack access point can directly transmit data to another wireless access point. Moreover, since the directional antenna is adopted, the interference range of each access point is limited by its transmission direction [5]. The capacity of each link is set as 1Gbps (i.e., $C^F_{s_i s_j} = C^W_{a_x a_y} = 1G$, $\forall e^F_{s_i s_j} \in \mathbb{E}^F$, $e^W a_x a_y \in \mathbb{E}^W$). We consider two multicast groups in this example. For the first multicast group, the sender is placed in rack 1; the set of destinations includes racks 9, 10, 11, and 12; and the data rate of the multicast group is set as 1Gbps. For the second multicast group, the sender is set as rack 4; the set of destinations includes rack 5, 6, 7, and 8; and the data rate of the multicast group is 1Gbps. Now, we have to build a multicast tree, comprised of wired and wireless links, for each multicast group.

As shown in Fig. 2(a), we only adopt wired links to build multicast trees as it is for traditional data centers. In this case, the senders of top-of-rack switch 1 and 4 first transmit multicast data to the aggregation switch. Then, the aggregation switch has to transmit the same multicast data through four different wired links for the four destinations. For the two multicast trees, the total number of links used is 10 and the total multicast data traffic is 10×1 Gbps $= 10$ Gbps. We can see that the multicast trees with purely wired links result in severe data redundancy. In Fig. 2(b), when the wireless access points are considered, the multicast data of the first multicast group can be transmitted by the access point of rack 1 to that of rack 9. Then, the wireless access point of rack 9 transmits data to the access point of rack 12. Thus, rack 10, 11, and 12 can simultaneously receive the multicast data. This multicast tree only uses the two wireless links. For the second multicast group, since the access point of rack 5 is interfered by the wireless transmission of the access point on rack 1, the multicast data is selected to be transmitted by the wired links and occupies five wired links. The total multicast data traffic of the two multicast trees is 7×1 Gbps=7 Gbps.

Actually, we have a better option to build the multicast trees as shown in Fig. 2(c). Interestingly, we can utilize the wired links to avoid wireless interference such that more



(a) (b) (c)

FIGURE 2. An illustrative example for multicast tree construction in wireless data centers. (a) Multicast tree construction I. (b) Multicast tree construction II. (c) Multicast tree construction III.

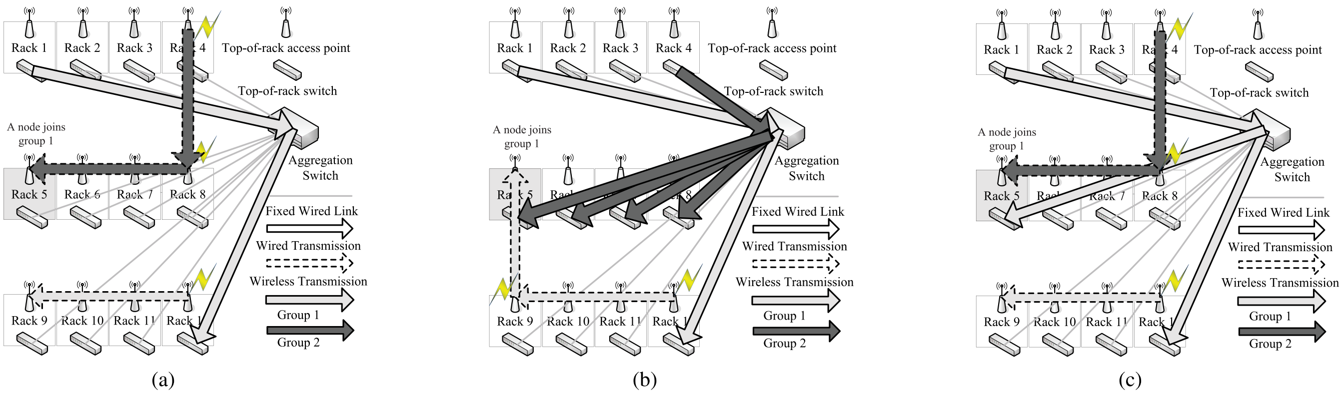


FIGURE 3. **An illustrative example for multicast tree maintenance in wireless data centers. (a) Multicast tree construction III. (b) Multicast tree maintenance I. (c) Multicast tree maintenance II.**

wireless access points can be simultaneously transmitted to further reduce the data redundancy. The data of the first multicast group can pass through the aggregation switch from rack 1 to rack 12. Then, the wireless access point of rack 12 can relay the data to rack 9. The multicast tree for the first multicast group is comprised of two wired links and one wireless link. Then, the multicast data of the second multicast group can be transmitted by the two wireless access points on rack 4 and 8. The total data traffic for the two group communications is $(3 + 2) \times 1$ Gbps $= 5$ Gbps.

2) MULTICAST TREE MAINTENANCE

The example in Fig. 3 depicts the multicast tree maintenance problem, where the same system settings are used as that in the example of the multicast tree construction. Moreover, in this example, we adopt the two multicast trees constructed in the example of multicast tree construction and consider that a node of rack 5 joins multicast group 1, as shown in Fig. 3(a). Then, we attempt to maintain the multicast trees such that the node can receive the multicast data. As shown in Fig. 3(b), the involved group (i.e., multicast group 1) intuitively uses the wireless link to relay data from rack 9 to rack 5. However, because the transmission interferes the wireless transmission of the access point on rack 8, the multicast data of the victim group (i.e., group 2) is forced to be delivered via the wired links. As a result, totally 4 Gbps redundant multicast data traffic is increased. However, in this case, we should use the wired link to transmit the data of group 1 instead. The data can then pass through the aggregation switch from rack 1 to rack 5 as shown in Fig. 3(c) and we only have 1 Gbps more redundant data traffic under this solution. This example demonstrates that the tree maintenance problem is important and nontrivial in the minimization of the multicast data traffic and has to be carefully addressed.

IV. THE MULTICAST TREE CONSTRUCTION

In this section, we prove the $\mathcal{NP}$-hardness of the problem by a reduction from the *partition problem*, which is known to be $\mathcal{NP}$-complete [23], and propose an efficient heuristic algorithm to solve the multicast tree construction problem.

A. PROBLEM HARDNESS

Theorem 1: The multicast tree construction problem is $\mathcal{NP}$-hard.

Proof: The input instance of the partition problem is a set of M integers, $\mathbb{B} = \{b_1, b_2, \ldots, b_M\}$. The output is *YES* if and only if $\mathbb{B}$ can be partitioned into two subsets $\mathbb{U}$ and $\mathbb{B} \backslash \mathbb{U}$ with the same sum, i.e., $\sum_{b_m \in \mathbb{U}} b_m = \sum_{b_m \notin \mathbb{U}} b_m = \frac{1}{2} \sum_{b_m \in \mathbb{B}} b_m$.

Given an instance $\langle \mathbb{B} \rangle$ of the partition problem, we explain how to construct an instance $\langle \mathbb{G}, C^F_{s_i s_j}, C^W_{a_x a_y}, \mathbb{R}, N \rangle$ of our problem in polynomial time such that $\mathbb{B}$ can be evenly partitioned if and only if there exist M multicast trees with total data traffic $\frac{3}{2} \sum_{b_m \in \mathbb{B}} b_m$. The construction is as follows: We consider the wireless data center structure $\mathbb{G}$ shown in Fig. 1. There are twelve racks, each of which is equipped with a top-of-rack switch and a top-of-rack access point (i.e., $|\mathbb{V}^F| = 12$ and $|\mathbb{V}^W| = 12$). The capacity of each wired and wireless link is set at $\frac{1}{2} \sum_{b_m \in \mathbb{B}} b_m$ (i.e, $C^F_{s_i s_j} = C^F_{s_j s_i} = C^W_{a_x a_y} = C^W_{a_y a_x} = \frac{1}{2} \sum_{b_m \in \mathbb{B}} b_m$. There is a set of M multicast groups (i.e., $N = M$). The multicast data of M multicast groups is transmitted from rack 1 (source) to rack 5 (destination) (i.e., $\nu_m = 1$ and $\mathbb{D}_m = 5, \forall 1 \leq m \leq M$). The data rate of multicast group m is set as $T_m = b_m$, $\forall 1 \leq m \leq M$.

To complete the proof, we show that two partitioned subsets can be used to derive M multicast trees whose total data traffic is $\frac{3}{2} \sum_{b_m \in \mathbb{B}} b_m$, and vice versa. If there are two partitioned subsets, each integer b_m corresponds to the data rate T_m required by multicast group m. A subset corresponds to the data rate of the multicast groups transmitted by the two wired links (i.e., the wired switch of rack 1 to the aggregation switch and the aggregation to the wired switch of rack 5). The other subset corresponds to the data rate of the other multicast groups directly transmitted by the wireless link (i.e., the access point of rank 1 to the access point of rack 5).

IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING

Thus, the three links respectively transmit the data rate of $\frac{1}{2}\sum_{b_m \in \mathbb{B}} b_m$ and the total data traffic of M multicast trees is $\frac{3}{2}\sum_{b_m \in \mathbb{B}} b_m$. On the other hand, if the total data traffic of M multicast trees is $\frac{3}{2}\sum_{b_m \in \mathbb{B}} b_m$, the two wired links and the wireless link have to respectively transmit the data rate of $\frac{1}{2}\sum_{b_m \in \mathbb{B}} b_m$. It implies that the set can be evenly partitioned by assigning the corresponding integers into the corresponding subset. The existence of a polynomial-time algorithm for the partition problem implies the same for ours, which completes the proof. □

B. ALGORITHM DESCRIPTION

In this section, we propose an efficient algorithm for building multicast trees, comprised of wired and wireless links, for all multicast groups. The concept of this algorithm is to find some wireless access points that can cover as more destinations as possible to reduce the data redundancy of multicast traffic. Then, we find shortest paths, comprised of wired and wireless links, to connect each source with its destinations. Moreover, in order to use as few number of links as possible, for each shortest path, we will try to use wireless links first. If the wireless link cannot support the data transmission, we will utilize the wired link instead. Moreover, in order to efficiently utilize each link capacity, we will give a higher priority for the multicast group with a higher data rate to construct the multicast tree.

The pseudo-code of the proposed algorithm is shown in Algorithm 1. In Line 1, an indicator function $l^F(k, e^F_{s_is_j})$ is used to record whether wired link $e^F_{s_is_j}$ is allocated for transmitting the data of multicast group k, and is initialized as 0, $\forall 1 \leq k \leq N, e^F_{s_is_j} \in \mathbb{E}^F$. In Line 2, an indicator function $l^W(k, e^W_{a_xa_y})$ is used to record whether wireless link $e^W_{a_xa_y}$ is allocated to transmit the data of multicast group k, and is initialized as 0, $\forall 1 \leq k \leq N, e^W_{a_xa_y} \in \mathbb{E}^W$. In Line 3, a variable P_k, initialized as 0, is used record the priority of multicast group k. If multicast group k has a higher value of P_k, we have a higher priority to build a multicast tree for the multicast group. In Line 4, a set $\hat{\mathbb{E}}^W_k$ is used to record which wireless links can be adopted for delivering the traffic of multicast group k. In Line 5, a set $\hat{\mathbb{S}}^W_k$ is adopted to record how many destinations of multicast group k can overhear the multicast data transmitted by the access points of the destinations (racks). In Line 6, a set $\hat{\mathbb{D}}_k$ is used to register which destinations of multicast group k can receive the data and initialized as $\emptyset$.

Then, the algorithm starts to construct a multicast tree, comprised of wireless and wired links, for each multicast group (Lines 7-29). For each multicast group k, since the directional antenna with narrow-beam is generally adopted by wireless data centers, we let each wireless access point a_x, $\forall x \in \mathbb{D}_k \bigcup v_k$, attempt to transmit the data of multicast group k to each wireless access point a_y, $\forall y \in \mathbb{D}_k \bigcup v_k$, and compute how many destinations can receive the data (Lines 7-13). In Lines 10-11, if access point a_x of rack x can transmit the data to access point a_y of rack y

Algorithm 1 Multicast Tree Construction

Input: $\mathbb{G}, C^F_{s_is_j}, C^W_{a_xa_y}, \mathbb{R}, N$
Output: $l^F(k, e^F_{s_is_j}), l^W(k, e^W_{a_xa_y})$
1: $l^F(k, e^F_{s_is_j}) \leftarrow 0, \forall 1 \leq k \leq N, e^F_{s_is_j} \in \mathbb{E}^F$
2: $l^W(k, e^W_{a_xa_y}) \leftarrow 0, \forall 1 \leq k \leq N, e^W_{a_xa_y} \in \mathbb{E}^W$
3: $P_k \leftarrow 0, \forall 1 \leq k \leq N$
4: $\hat{\mathbb{E}}^W_k \leftarrow \emptyset, 1 \leq k \leq N$
5: $\hat{\mathbb{S}}^W_k \leftarrow \emptyset, \forall 1 \leq k \leq N$
6: $\hat{\mathbb{D}}_k \leftarrow \emptyset, \forall 1 \leq k \leq N$
7: **for** $k = 1$ to N **do**
8:   **for all** $x \in (\mathbb{D}_k \bigcup v_k)$ **do**
9:     **for all** $y \in (\mathbb{D}_k \bigcup v_k)$ **do**
10:       **if** $e^W_{a_xa_y} \in \mathbb{E}^W$ **then**
11:         $\hat{\mathbb{S}}^W_k \leftarrow \hat{\mathbb{S}}^W_k \bigcup (\mathbb{S}^W_{a_xa_y} \bigcap \mathbb{D}_k)$
12:         $\hat{\mathbb{E}}^W_k \leftarrow \hat{\mathbb{E}}^W_k \bigcup e^W_{a_xa_y}$
13:   $P_k \leftarrow T_k \times |\hat{\mathbb{S}}^W_k|$
14: Re-arrange the multicast group indexes by decreasing the priority of $P_k, \forall 1 \leq k \leq N$, such that $P_1 \geq P_2 \cdots \geq P_N$
15: **for** $k = 1$ to N **do**
16:   Re-arrange the wireless link indexes by decreasing the $(\mathbb{S}^W_{a_xa_y} \bigcap \mathbb{D}_k), \forall e^W_{a_xa_y} \in \hat{\mathbb{E}}^W_k$
17:   **for all** $e^W_{a_xa_y} \in \hat{\mathbb{E}}^W_k$ **do**
18:     **if** the access point capability constraint is satisfied and $|\mathbb{D}_k \bigcap \mathbb{S}_{a_xa_y}| \geq 2$ and $\hat{\mathbb{D}}_k \bigcap \mathbb{S}_{a_xa_y} = \emptyset$ **then**
19:       $\hat{\mathbb{D}}_k \leftarrow \hat{\mathbb{D}}_k \bigcup x$
20:       $l^W(k, e^W_{a_xa_y}) \leftarrow 1$
21:       SHORTEST-PATH(v_k, x)
22:       **for all** $v \in \mathbb{D}_k \bigcap \mathbb{S}_{a_xa_y}$ **do**
23:         **if** the access point capability constraint is satisfied **then**
24:           $\hat{\mathbb{D}}_k \leftarrow \hat{\mathbb{D}}_k \bigcup v$
25:         **else**
26:           Build a shortest path by wired links from v_k to v and set corresponding $l^F(k, e^F_{s_is_j})$ as 1
27:           $\hat{\mathbb{D}}_k \leftarrow \hat{\mathbb{D}}_k \bigcup v$
28:   **if** $\mathbb{D}_k \setminus \hat{\mathbb{D}}_k \neq \emptyset$ **then**
29:     SHORTEST-PATH$(v_k, \mathbb{D}_k \setminus \hat{\mathbb{D}}_k)$
30: **return** $l^W(k, e^W_{a_xa_y})$ and $l^F(k, e^F_{s_is_j}), \forall\ e^W_{a_xa_y}, e^F_{s_is_j}$

(i.e., $e^W_{a_xa_y} \in \mathbb{E}^W$), a set of destinations can receive the data (*i.e.*, $\mathbb{S}^W_{a_xa_y} \bigcap \mathbb{D}_k$); and the set $\hat{\mathbb{S}}^W_k$ is updated to $\hat{\mathbb{S}}^W_k \bigcup (\mathbb{S}^W_{a_xa_y} \bigcap \mathbb{D}_k)$. In Line 12, the wireless link $e^W_{a_xa_y}$ that can be used for transmitting the data of multicast group k is added into the set $\hat{\mathbb{E}}^W_k$. When all pairs of the access points of destinations are tried out, the priority P_k of multicast group k is set as $T_k \times |\hat{\mathbb{S}}^W_k|$ (Line 13). That is, if more destinations can overhear the data transmitted by the wireless access points and the traffic of multicast group k has a higher data rate, more data redundancy can be reduced. Thus, we give a higher priority for the multicast group to build multicast tree and to use wireless access points.

After the priorities of all multicast groups are set, we re-arrange the multicast group indexes by decreasing the priority of P_k, $\forall 1 \le k \le N$, such that $P_1 \ge P_2 \cdots \ge P_N$ (Line 14). Then, we start to build a multicast tree for each multicast group and adopt the new index of multcast group, i.e., multicast group $k = 1$ has the highest property P_1 (Lines 15-29). For multicast group k, we re-arrange the wireless link indexes $e^W_{a_x a_y} \in \hat{\mathbb{E}}^W_k$ by decreasing the $(\mathbb{S}^W_{a_x a_y} \bigcap \mathbb{D}_k)$ in order to select the wireless links covering as more destinations as possible (Line 16). Then, for each wireless link $e^W_{a_x a_y} \in \hat{\mathbb{E}}^W_k$, we select access point a_x transmitting data to access point a_y if the following three conditions are met (Lines 17-18): 1) the access point can meet its capability constraint; 2) at least two destinations can simultaneously receive the multicast data (i.e., $|\mathbb{D}_k \bigcap \mathbb{S}_{a_x a_y}| \ge 2$); and 3) each destination of multicast group k cannot receive the same multicast data from more than one link in order to meet the tree properties (i.e., $\hat{\mathbb{D}}_k \bigcap \mathbb{S}_{a_x a_y} = \emptyset$). If the link is adopted, we add destination (rack) x, which can receive data, to the registered destination set $\hat{\mathbb{D}}_k$ (i.e., $\hat{\mathbb{D}}_k = \hat{\mathbb{D}}_k \bigcup x$) (Line 19) and the indicator function $l^W(k, e^W_{a_x a_y})$ is set as 1 accordingly (Line 20). Although the wireless link $e^W_{a_x a_y}$ is adopted and can transmit data to some destinations, access point a_x does not have a path to receive the multicast traffic from sender v_k. Then, we find a shortest path, comprised of wired and wireless links, for the given pair of source v_k and access point a_x of rack x. Whenever Procedure SHORTEST-PATH() is invoked, it attempts to find a shortest path from source v_k of multicast group k to destination x through as few links as possible (Line 21). For the path, we try to use wireless links first. If the wireless links do not satisfy the access point capability constraint, we adopt wired links instead. Then, the corresponding indicator functions $l^W(k, e^W_{\tilde{x}\tilde{y}})$ and $l^F(k, e^F_{ij})$ are set as 1.

In Lines 22-27, although the access point a_v of the destination rack v can overhear the wireless transmission (i.e., $v \in \mathbb{D}_k \bigcap \mathbb{S}_{a_x a_y}$), it may not have enough capability to receive the data. Therefore, if the access point has capability to receive the data, we directly add the destination of rack v to the registered destination set $\hat{\mathbb{D}}_k$ (Line 24). Otherwise, we build a shortest path by wired links from sender v_k to destination v and set corresponding $l^F(k, e^F_{s_i s_j})$ as 1 (Line 26). The destination of rack v is also added to the registered destination set $\hat{\mathbb{D}}_k$ (Line 27). Finally, if there are some remaining destinations that have no path to receive multicast data (i.e., $\mathbb{D}_k \setminus \hat{\mathbb{D}}_k \neq \emptyset$), we use Procedure SHORTEST-PATH() to find a shortest path for each remaining destination of multicast group k (Lines 28-29). Finally, we return a multicast tree, comprised of wireless and wired links, for each multicast group (Line 30).

Theorem 2: The time complexity of Algorithm 1 is $O(N\tilde{D}(\tilde{E}\omega + \tilde{D}))$. $\tilde{D} = \max_{\forall k} |\mathbb{D}_k|$; $\tilde{E} = \max(|\mathbb{E}^W|, |\mathbb{E}^F|)$. ω is the running time of the shortest path algorithm.

Proof: The initialization process requires $O(N\tilde{E})$ time. For each multicast group k, a priority P_k is computed only once and can be done in $O(\tilde{D}^2)$. Thus, for N multicast groups, the algorithm takes $O(N\tilde{D}^2)$ time. For building a multicast tree of group k, there are at most $\tilde{D}$ destinations and $\tilde{E}$ links; and Procedure SHORTEST-PATH() is used only once for each destination. Building multicast trees for N multicast groups takes $O(N\tilde{E}\tilde{D}\omega)$. Thus, the time complexity of Algorithm 1 is $O(N\tilde{D}(\tilde{E}\omega + \tilde{D}))$. □

V. THE MULTICAST TREE MAINTENANCE

In this section, we also show that the problem is $\mathcal{NP}$-hard, and respectively propose an efficient heuristic algorithm to maintain the multicast trees for nodes joining and leaving.

A. PROBLEM HARDNESS

Theorem 3: The multicast tree maintenance problem is $\mathcal{NP}$-hard.

Proof: This theorem can be proved in a similar way to Theorem 1. The input instance in Theorem 1 is reused in this theorem. We describe how to construct the additional inputs of the multicast tree maintenance problem (i.e., J_m and L_m). Any M multicast trees have been constructed in the multicast tree construction problem and the capacity of each wired and wireless link is exhausted. Now, we consider that rack 5 and rack 9 are additionally equipped with one wired switch and connected with two wired links so that the two racks can transmit data directly. Each multicast group m has a node in rack 9 requesting to join (i.e., $|J_m| = 1$) and does not have any node requesting to leave (i.e., $|L_m| = 0$). The multicast data of M multicast groups also has to transmit to rack 9 (destination) from rack 5 (i.e., $\mathbb{D}_m = \mathbb{D}_m \bigcup 9$, $\forall 1 \le m \le M$).

To complete the proof, we show that two partitioned subsets can be used to derive the tree maintenance for M multicast trees whose the increased data traffic is $\sum_{b_m \in \mathbb{B}} b_m$, and vice versa. If there are two partitioned subsets, each integer b_m corresponds to the data rate T_m required by multicast group m. A subset corresponds to the data rate of the multicast groups. The data of the multicast groups is directly transmitted via one wired link. The other subset corresponds to the data rate of the other multicast groups, which should be transmitted by the other wired link. Since each wired link transmits the data rate of $\frac{1}{2}\sum_{b_m \in \mathbb{B}} b_m$, the totally increased data traffic is $\sum_{b_m \in \mathbb{B}} b_m$. On the other hand, if the totally increased data traffic of M multicast trees is $\sum_{b_m \in \mathbb{B}} b_m$, each wired link has to respectively transmit the data rate of $\frac{1}{2}\sum_{b_m \in \mathbb{B}} b_m$. It implies that the set can be evenly partitioned by assigning the corresponding integers into the corresponding subset. The existence of a polynomial-time algorithm for the partition problem implies the same for ours, which completes the proof. □

B. ALGORITHM DESCRIPTION FOR NODE JOINING

This section propose a polynomial time algorithm to deal with the multicast tree maintenance problem for node joining. When there are nodes requesting to join multicast groups, how to maintain each multicast tree is a complicated problem.

Algorithm 2 Node Joining

Input: $\mathbb{G}, C^F_{s_is_j}, C^W_{a_xa_y}, J_k, N, l^F(k, e^F_{s_is_j}), l^W(k, e^W_{a_xa_y}), \mathbb{R}$

1: $\hat{l}^F(k, e^F_{a_xa_y}) \leftarrow l^W(k, e^W_{a_xa_y}), \forall 1 \leq k \leq N, e^F_{s_is_j} \in \mathbb{E}^F$

2: $\hat{l}^F(k, e^F_{s_is_j}) \leftarrow l^F(k, e^F_{s_is_j}), \forall 1 \leq k \leq N, e^W_{a_xa_y} \in \mathbb{E}^W$

3: **for** $k = 1$ to N **do**

4:   **for all** $\mathrm{j}_k \in J_k$ **do**

5:     Flag = false

6:     **for all** $\{e^W_{a_xa_y} | l^W(k, e^W_{a_xa_y}) = 1\}$ **do**

7:       **if** $\mathrm{j}_k \in \mathbb{S}_{a_xa_y}$ **then**

8:         Flag = true

9:         **break**

10:     **if** Flag = false **then**

11:       **for all** $\{e^W_{a_xa_y} | l^W(k, e^W_{a_xa_y}) = 1\}$ **do**

12:         **if** $e^W_{a_xa_{\mathrm{j}_k}} \in E^W$ **then**

13:           CHECK-CAPABILITY$(e^W_{a_xa_y}, e^W_{a_xa_{\mathrm{j}_k}})$

14:           Flag = true

15:           **break**

16:         **else if** $e^W_{a_{\mathrm{j}_k}a_y} \in E^W$ **then**

17:           CHECK-CAPABILITY$(e^W_{a_xa_y}, e^W_{a_{\mathrm{j}_k}a_y})$

18:           Flag = true

19:           **break**

20:     **if** Flag = false **then**

21:       SHORTEST-PATH(v_k, j_k)

22: **return** $\hat{l}^W(k, e^W_{a_xa_y})$ and $\hat{l}^F(k, e^F_{s_is_j}), \forall\, e^W_{a_xa_y}, e^F_{s_is_j}$

Specifically, when a node joins a multicast group in a rack and we would like to transmit data to the rack via a wireless link, multiple wireless links of the existed groups may interfere the access point of the rack. Under the limited capacity of the access point, some groups have to change their tree structures as the victim groups. However, each victim group has tremendous choices to select other substitute paths via wired and/or wireless transmissions. With the consideration of the feasibility, we are impossible to process all the possible selections in our algorithm. To tackle this problem, we design a procedure, named *collision procedure*, by observing the structure of the wireless data centers to sieve out an efficient substitute path from all the possible selections. In the procedure, we build the substitute path for each victim group and avoid the chain reaction, when the victim groups have to change their tree structures.

The pseudo-code of the algorithm is shown in Algorithm 2. In Lines 1-2, the new indicator function $\hat{l}^W(k, e^W_{a_xa_y})$ and $\hat{l}^F(k, e^W_{s_is_j})$ are initially set as the wired and wireless links of the multicast trees constructed in Algorithm 1. Then, the algorithm starts to reconstruct multicast trees for the joining requests (Lines 3-21). For rack j_k, "Flag", initialled as false, is used to indicate that rack j_k can receive the multicast data of group k or not (Line 5). Then, we check whether the rack is covered by a wireless link of itself tree structure and directly receive the data. It implies the tree structure of group k does not require to be changed and Flag is set as true (Lines 6-9). Otherwise, we attempt to adjust the tree such that the rack can receive the data (Lines 10-19). We try to lengthen each wireless link which is already used by group k and there are two possible directions (Lines 11-19). For each used wireless link $e^W_{a_xa_y}$ of group k, the first case for the lengthened direction is rack j_k as the new destination in the right hand side of the original destination (i.e, rack y) and the wireless link $e^W_{a_xa_y}$ is changed as $e^W_{a_xa_{\mathrm{j}_k}}$ (Lines 12-13). The other case is rack j_k as the new sender in the left hand side of the original sender (i.e., rack x) and the wireless link $e^W_{a_xa_y}$ is changed as $e^W_{a_{\mathrm{j}_k}a_y}$ (Lines 16-17). Since the lengthened wireless link will interfere more access points on the racks such that their capacity may not be sufficient (abbreviated as *collision racks*), it implies that many wireless links of other groups, which pass through the collision racks, will be affected as well. Therefore, Procedure CHECK-CAPABILITY() is involved to check the capacity of each access point, covered by the lengthened wireless, link and determine which groups should be the victim groups to change their tree structures (Line 13 or 17). If we cannot transmit data to rack j_k by lengthening a wireless link from the original multicast tree, we build a shortest path with wired links to transmit data to rack j_k by involving Procedure SHORTEST-PATH() (Lines 20-21).

Procedure CHECK-CAPABILITY$(e^W_{a_xa_y}, e^W_{a_ua_t})$

1: **for all** $a \in \mathbb{S}_{a_ua_t}$ **do**

2:   **if** the capability constraint of access point a is not satisfied **then**

3:     $\hat{B}_k \leftarrow \hat{B}_k \bigcup a$

4: **if** $|\hat{B}_k| = \emptyset$ **then**

5:   $\hat{l}^W(k, e^W_{a_xa_y}) \leftarrow 0$ and $\hat{l}^W(k, e^W_{a_ua_t}) \leftarrow 1$

6: **else**

7:   COLLISION$(\hat{B}_k)$

Procedure CHECK-CAPABILITY() takes original wireless link $e^W_{a_xa_y}$ and lengthened wireless link $e^W_{a_ua_t}$ as inputs. When lengthened wireless link $e^W_{a_ua_t}$ is used, each access point $a \in \mathbb{S}_{a_ua_t}$ will be interfered. If the capability constraint of an access point $a \in \mathbb{S}_{a_ua_t}$ is not satisfied, we add the access point of the rack to set $\hat{B}_k$ (Lines 1-3). If the capacity constraint of all the access points are satisfied (i.e., $|\hat{B}_k| = 0$), lengthened wireless link $e^W_{a_ua_t}$ is adopted (i.e., $\hat{l}^W(k, e^W_{a_ua_t}) = 1$) and original wireless link $e^W_{a_xa_y}$ is released (i.e., $\hat{l}^W(k, e^W_{a_xa_y}) = 0$) (Lines 4-5). Otherwise, we trigger Procedure COLLISION() to determine which groups, with wireless links passing through the collision rack, should be the victim groups to change their tree structures.

Procedure COLLISION() (see next page) takes the set of collision racks $\hat{B}_k$ as input. This procedure is to determine which groups should be the victim groups to change their tree structure. If there is only one collision rack (i.e., $|\hat{B}_k| = 1$), we calculate a priority $\hat{P}_g$, initialized as 0, for each multicast group $g \in \mathbb{M}_{\hat{B}_k}$ (Line 1), where $\mathbb{M}_{\hat{B}_k}$ is the set of groups which has a wireless link passing through the access point of the collision rack (Lines 1-7). The higher the priority,

Procedure COLLISION($\hat{B}_k$)

1: $\hat{P}_g \leftarrow 0, \forall 1 \leq g \leq N$
2: **if** $|\hat{B}_k| = 1$ **then**
3:   **for all** $g \in \mathbb{M}_{\hat{B}_k}$ **do**
4:     **for all** $\{e^W_{a_x a_y} | \hat{l}^W(g, e^W_{a_x a_y}) = 1\}$ **do**
5:       **if** $\hat{B}_k \in \mathbb{S}_{a_x a_y}$ **then**
6:         $\hat{P}_g = \{\text{HOPPING}() - T_g\}$
7:   Re-arrange the wireless link indexes by decreasing the priority of $\hat{P}_g$
8:   **for all** $g \in \mathbb{M}_{\hat{B}_k}$ **do**
9:     **if** $\hat{P}_g > 0$ and the capacity constraint of the access point $\hat{B}_k$ is not satisfied **then**
10:       set the corresponding wireless and wired link indicator function as 1
11: **else**
12:   Build a shortest path by wired links from ν_k to j_k and set corresponding $\hat{l}^F(k, e^F_{s_i s_j})$ as 1

the more the increased data redundancy. Then, according to the priorities, the groups with higher priorities will still use the original wireless link. Until the capacity of the access point is not enough, the other groups with lower priorities will be the victims groups to change their paths. Otherwise, if the collision racks are more than one, with the consideration of the feasibility for computation complexity, we will use wired links to connect the joining node in the rack j_k. (Lines 11-12).

Now, we explain how to calculate priority P_g for group g (Line 6). If the wireless link of group g, passing through the collision rack, is released, we have to rebuild a path instead of the released wireless link. For finding a substitute path, we are impossible to search all the possible paths. Thus, we observe the structure of the wireless data center to find an efficient substitute path comprised of wired and wireless links as shown in Fig. 4, when the group should be the victim group to change its tree structure.

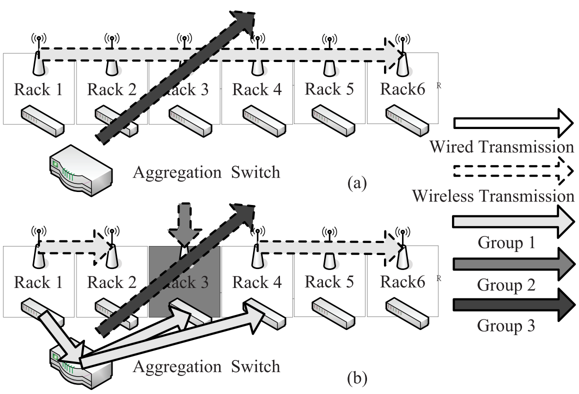


FIGURE 4. An illustration for Procedure COLLISION(). (a) The original wireless links. (b) Group 1 is the victim group when a node joins group 2 in rack 3.

Fig. 4(a) shows a wireless link of group 1 and 3 when no any node requests to join. When a node requests to join group 2 in rack 3, a wireless link is lengthened to rack 3 for transmitting the data to the node such that the capacity of the access point on rack 3 is not enough. Let group 1 be the victim group. Then, we rebuild a substitute path, comprised of two wireless links and three wired links, for the destinations of group 1 in order to avoid the interference on the access point on rack 3, as shown in Fig. 4(b). Thus, for the new path of group 1, the priority (increased data redundancy) P_1 is $5T_1$ - T_1, where $5T_1$ is the data redundancy of group 1 under the new substitute path in Fig. 4(b) and T_1 is the data redundancy of group 1 under the original wireless link in Fig. 4(a). Priority P_g will be calculated by Function Hopping(). Consequently, groups with low P_g will be the victim groups in order to reduce the increased data redundancy and we set the corresponding wired and wireless link indicator function as 1 for the new substitute path (Lines 8-10).

Theorem 4: The time complexity of Algorithm 2 is $O(N\tilde{J}(\tilde{S}\tilde{E} + \tilde{M}\tilde{E}^2 + \omega))$. $\tilde{J} = \max_{\forall k} |J_k|$; $\tilde{S} = \max_{\forall a_x a_y}(|\mathbb{S}_{a_x a_y}|)$; $\tilde{M} = \max_{\forall k} |\mathbb{M}_{\hat{B}_k}|$.

Proof: There are at most N groups (Line 3 of Algorithm 2). For each multicast group k, at most $\tilde{J}$ racks have to receive data of group k (Line 4 of Algorithm 2). For each rack which has nodes joining to group k, we try to lengthen a wireless link selected from at most $\tilde{E}$ wireless links to transmit data to the rack (Lines 6-19 of Algorithm 2). If we can lengthen a wireless link to transmit data to the rack, Procedure CHECK-CAPACITY() and COLLISION() will be involved (Lines 10-19 of Algorithm 2). Procedure CHECK-CAPACITY() will check the capacity of the access points covered by the lengthened wireless link and takes $O(\tilde{S})$ time (Lines 1-3 of Procedure CHECK-CAPACITY). Procedure COLLISION() will compute a priority for each group which has a wireless link passing through the collision rack. Since there are at most $\tilde{M}$ groups each of which has at most $\tilde{E}$ wireless links to be checked, this procedure takes $O(\tilde{M}\tilde{E})$ time (Lines 3-6 of Procedure COLLISION). Since there are at most $\tilde{E}$ wireless links, searching wireless links for N groups, each of which has nodes joining in at most $\tilde{J}$ racks, takes $O(N\tilde{J}(\tilde{S}\tilde{E} + \tilde{M}\tilde{E}^2))$ time. Otherwise, if no any wireless link is suitable for transmitting data to the rack, wired links via involving SHORTEST-PATH() which takes $O(\omega)$ time, are instead. Thus, the time complexity of Algorithm 2 is $O(N\tilde{J}(\tilde{S}\tilde{E} + \tilde{M}\tilde{E}^2 + \omega))$ (Lines 3-21 of Algorithm 2). ☐

C. ALGORITHM DESCRIPTION FOR NODE LEAVING

In this section, we propose a polynomial time algorithm to maintain the multicast trees for node leaving. The concept of the algorithm design is to retrieve unused wireless links and reassign the wireless resource to other groups. When a node leaves a multicast group, the wireless resource on a rack could be released and the released resource can be used for other multicast groups which use wired links to transmit data to the rack. Since multiple groups on the rack have to compete the wireless resource, we have to determine which groups should

Algorithm 3 Node Leaving

Input $\mathbb{G}$, $C^F_{s_is_j}$, $C^W_{a_xa_y}$, L_k, N, $l^F(k, e^F_{s_is_j})$, $l^W(k, e^W_{a_xa_y})$, $\mathbb{R}$

1: $\hat{l}^F(k, e^F_{s_is_j}) \leftarrow \hat{l}^W(k, e^W_{a_xa_y}), \forall 1 \leq k \leq N, e^F_{s_is_j} \in \mathbb{E}^F$
2: $\hat{l}^F(k, e^F_{s_is_j}) \leftarrow l^F(k, e^F_{s_is_j}), \forall 1 \leq k \leq N, e^W_{a_xa_y} \in \mathbb{E}^W$
3: **for** $k = 1$ to N **do**
4:   **for all** $l_k \in L_k$ **do**
5:     **for all** $\{e^W_{a_xa_y} | l^W(k, e^W_{a_xa_y}) = 1\}$ **do**
6:       **if** $l_k \in \mathbb{S}_{a_xa_y}$ and $\mathbb{D}_k \bigcap \mathbb{S}_{a_xa_y} = \emptyset$ and LeafNode(a_{l_k}) = true **then**
7:         $l^W(k, e^W_{a_xa_y}) \leftarrow 0$
8:         REALLOCATION($\mathbb{S}_{a_xa_y}$)
9:         PRUNE($a_x, \mathbb{D}_k$)
10:         **break**
11: **return** $\hat{l}^W(k, e^W_{a_xa_y})$ and $\hat{l}^F(k, e^F_{s_is_j}), \forall e^W_{a_xa_y} \in \mathbb{E}^W, e^F_{s_is_j} \in \mathbb{E}^F$

use the wireless resource instead of wired links and how to use. Moreover, when there are nodes requesting to join, Algorithm 2 may generate some victim groups and rebuilds a substitute path for the victim groups. We also address how to recover an efficient path from the substitute path. To deal with the above problems, we respectively design a procedure *prune* and *reallocation* to retrieve unused wireless links and reassign the released wireless resource to other groups.

The pseudo-code of the proposed algorithm for node leaving is shown in Algorithm 3. In Lines 1-2, the new indicator functions $\hat{l}^W(k, e^W_{a_xa_y})$ and $\hat{l}^F(k, e^W_{s_is_j})$ are the same as the Lines 1-2 of Algorithm 2. For each leaving node $l_k \in L_k$ of group k, we check each wireless link used by group k whether can be retrieved if there are nodes of group k requesting to leave. The resource of a wireless link can be released when the following three conditions are met (Lines 4-6). 1) The leaving node is covered by the transmission range of the wireless link. 2) The transmission range of the wireless link does not cover any other destination. 3) The leaving node is a leaf node in the tree, because when the leaving node is not a leaf node, the wireless link may be used to relay data and cannot be released. If the resource of wireless link $e^W_{a_xa_y}$ can be released, we retrieve the wireless link and set the indicator function $l^W(k, e^W_{a_xa_y})$ as 0 (Line 7). Since the wireless link of group k is retrieved, the access points on the racks (abbreviated as "involved racks"), originally interfered by the wireless link, get free capacity T_k. Thus, Procedure REALLOCATION() is designed to reallocate the released wireless resource to other groups which use wired links to transmit data to the involved racks (i.e., $S_{a_xa_y}$) and determine which groups should use the released wireless resource instead of the wired links (Line 8). Because the leaving node is a leaf node of the tree, a path may include multiple wireless links to relay data to the leaf node from the root. Thus, we have chance to retrieve more wireless links of the path. Hence, Procedure PRUNE() tries to revoke more wireless links to further reduce data redundancy (Line 9). Finally, we return the two indicator functions (Line 11).

Procedure REALLOCATION($\mathbb{S}_{a_xa_y}$)

1: $\hat{P}_g \leftarrow 0, \forall 1 \leq g \leq N$
2: **for all** $z \in \mathbb{S}_{a_xa_y}$ **do**
3:   **for all** $g \in \mathbb{H}_z$ **do**
4:     LeftLink = false
5:     RightLink = false
6:     **for all** $\{e^W_{a_xa_y} | \hat{l}^W(g, e^W_{a_xa_y}) = 1\}$ **do**
7:       **if** $\hat{l}^W(g, e^W_{a_xa_z}) = 1$ and all access points capability are satisfied **then**
8:         LeftLink = true
9:       **else if** $\hat{l}^W(g, e^W_{a_za_y}) = 1$ and all access points capability are satisfied **then**
10:         RightLink = true
11:     **if** LeftLink = true and RightLink = true and the two wireless links can be combined **then**
12:       $\hat{P}_g$ = WIRED-COST($\hat{l}(g, e^F_{s_is_j})$) + T_g
13:     **else if** LeftLink = true or RightLink = true **then**
14:       $\hat{P}_g$ = WIRED-COST($\hat{l}(g, e^F_{s_is_j})$)
15:   Re-arrange the wireless link indexes by decreasing the priority of $\hat{P}_g$
16:   **for all** $g = 1$ to $|\mathbb{H}_z|$ **do**
17:     **if** $\hat{P}_g > 0$ and all access points capability constraint are satisfied **then**
18:       set the corresponding indicator function of wireless links as 1 and of wired links as 0

Procedure REALLOCATION() takes $\mathbb{S}_{a_xa_y}$ as input to reallocate wireless resource of each access point on each involved rack in $\mathbb{S}_{a_xa_y}$. In Line 1, variable $\hat{P}_g$, initialized as 0, is used to record a priority value for each multicast group. The value of $\hat{P}_g$ means an amount of the data redundancy used by group g. For each involved rack $z \in \mathbb{S}_{a_xa_y}$, there is a set of groups $\mathbb{H}_z$ which has a destination (node) in rack z and uses a wired link to transmit data to rack z (Line 2). For each group $g \in \mathbb{H}_z$, we attempt to lengthen an existed wireless link instead of the wired link to transmit data to the destination of group g in rack z (Lines 3-14). To lengthen each wireless link $e^W_{a_xa_y}$ which is already used by group k, there are two possible directions. The first one is that access point a_x can transmit data to rack z and rack z can be the new destination in the right hand side of rack y (i.e., $\hat{l}^W(g, e^W_{a_xa_z}) = 1$). If the wireless link can transmit data to rack z via lengthening, flag LeftLink is set as true (Lines 7-8). Similarly, the other one is that access point a_z, as the new sender in the left hand side of rack x, can transmit data to rack y (i.e., $\hat{l}^W(g, e^W_{a_za_y}) = 1$). If the access point on rack z can transmit data to rack y by lengthening the wireless link, flag RightLink is set as true (Lines 9-10).

Now, we calculate priority $\hat{P}_g$ for group g to record an amount of data redundancy that can be reduced. If the two flags are true and one of the two wireless links can cover all the destinations that the other wireless link can cover, it means that the two wireless links can combine as one wireless link. The priority of group g is set as WIRED-COST()+T_g (Lines 11-12), where WIRED-COST() will return that an amount of the wired link data rate used by the group g is retrieved and the value of T_g represents the retrieved wireless resource. Otherwise, if flag LeftLink or RightLink is set as true, the priority of group g is set as WIRED-COST() because only wired links can be retrieved (Lines 13-14). After each involved multicast group has a priority value, we re-arrange the involved group indexes by decreasing the priority of $\hat{P}_g$ (Line 15). Then, according to the priority value, the groups with higher priority will use the wireless resource first instead of the used wired links to reduce the data redundancy until the capacity of the access point on rack z is insufficient. Finally, we set the corresponding indicator function of wireless links as 1 and of wired links as 0 (Lines 16-18).

We use the same example shown in Fig. 4 to explain how to lengthen wireless links and to calculate a priority for group 1, when the node in rack 3 leaves group 2. For the destination of group 1 in rack 3, the first direction to lengthen a wireless link is that the access point on rack 1 can transmit data to rack 3. The other direction is that the access point on rack 3 as the new sender can transmit data to the access point on rack 6. Since the two wireless links can cover the same destinations, they can combine as one wireless link. Thus, the path of group 1 shown in Fig. 4(b) can recover to the original wireless link of group 1 shown in Fig. 4(a). Thus, group 1 only uses one wireless link instead of the five links. In this case, one wireless link and three wired links are retrieved. WIRED-COST() returns $3T_1$ and $\hat{P}_1$ is $4T_1$.

Procedure PRUNE(a_y, $\mathbb{D}_k$)
1: $a_x \leftarrow$ PARENT(a_y)
2: **if** LeafNode(a_y) = $\emptyset$ and $\mathbb{D}_k \bigcap \mathbb{S}_{a_x a_y} = \emptyset$ **then**
3: $\quad l^W(k, e^W_{a_x a_y}) \leftarrow 0$
4: $\quad$REALLOCATION($\mathbb{S}_{a_x a_y}$)
5: $\quad$PRUNE(a_x)

In Procedure PRUNE, we try to retrieve more wireless links of a path transmitting data to access point a_y. This is because a multicast tree may adopt many wireless links to relay data to only one destination. In Line 1, we use PARENT() to find the parent node of access point a_y. In order to ensure the connectivity of multicast tree, we retrieve the wireless link if the access point is a leaf node and the wireless link does not cover any other destination (Line 2). Then, we retrieve the wireless link and set the indicator function as 0 (Line 3). Then, since the resource of the wireless link is released, we trigger Procedure REALLOCATION() to reassign the wireless resource to other multicast groups which use wired links to transmit their multicast data (Line 4). In Line 5, we try to retrieve one more wireless link of the tree until the wireless link of next parent node cannot be revoked.

Theorem 5: The time complexity of Algorithm 3 is $O(N\tilde{L}\tilde{E}^2\tilde{S}\tilde{H}\tau)$. $\tilde{L} = \max_{\forall k}|L_k|$; $\tilde{H} = \max_{\forall z}(|\mathbb{H}_z|)$; $\tau = \max_{\forall k}(\text{TreeDepth}(k))$.

Proof: There are at most N groups (Line 3 of Algorithm 3). For each multicast group k, there are at most number of leaving nodes $\tilde{L}$ (Line 4 of Algorithm 3). For each leaving node, we attempt to retrieve a wireless link from at most $\tilde{E}$ wireless links (Line 5 of Algorithm 3). If a wireless link can be revoked, we reallocate the released wireless resource by involving Procedure REALLOCATION() and PRUNE() (Lines 6-9 of Algorithm 3). In Procedure REALLOCATION(), the number of the involved racks, covered by a wireless link, is at most $\tilde{S}$. For an involved rack, there are at most $\tilde{H}$ groups with a destination in the involved rack. For a group, we have to check at most $\tilde{E}$ wireless links and calculate a priority (Lines 2-14 of Procedure REALLOCATION()). The procedure takes $O(\tilde{S}\tilde{H}\tilde{E})$ time. In Procedure PRUNE(), if it retrieves a wireless link, Procedure REALLOCATION() will be involved once. Since the depth of a tree is at most τ, Procedure REALLOCATION() will be involved at most τ times. Thus, the complexity of Algorithm 3 is O($N\tilde{L}\tilde{E}^2\tilde{S}\tilde{H}\tau$). □

VI. PERFORMANCE EVALUATION

A. SIMULATION SETUPS

In this section, we develop a simulation model based on a realistic wireless data center topology, where the hierarchical topology is used according to the deployment of Microsoft [6], to evaluate our proposed algorithms. In the network architecture, there are 160 top-of-racks, each of which has one wired switch and one 60GHz wireless access point with a directional narrow-beam antenna. The real measurement results from Microsoft have indicated that two parallel 60GHz wireless links are interfered with each other when the distance of the two links is smaller than 22 inches. Note that the width of a rack is about 24 inches. By the geometric-based interference model and the deployment of wireless access points, the transmission range of each wireless link and its interference can be accordingly derived, and an example is shown in Fig. 5.

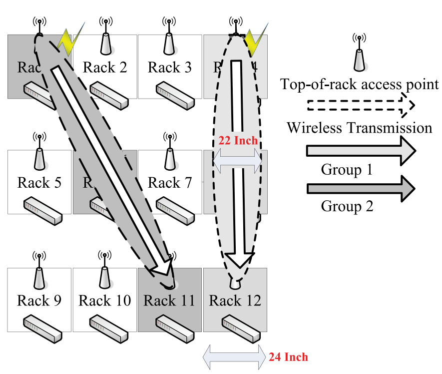


FIGURE 5. **An illustration for understanding the range of wireless interference in wireless data centers.**

The maximal capacity of each link is set as 1Gbps when background traffic is not considered. However, to investigate the impacts of background traffic, the available capacity of each link is randomly assigned from 300 Mbps to 1000 Mbps if background traffic is heavy in data centers [24]. On the other hand, for the case of light background traffic, the available capacity of each link is randomly set from 700 Mbps to 1000 Mbps. Moreover, the number of multicast groups in our experiments varies from 50 to 250 [13]. For each multicast group, one source and some destinations are randomly selected from 160 top-of-racks. To determine the number of destinations in a multicast group, we consider two different distributions [19]. The first one is uniform distribution with the range from 3 to 160. The other one is power-law distribution, which generates more small groups in the data center. The data rate for each multicast group is set based on the real data flows in a data center [25], and it is selected as one of the following six data rates, 1, 10, 100, 1000, 10000, 100000kbps, with the corresponding probabilities 0.1, 0.3, 0.2, 0.2, 0.15, and 0.05.

We compared our proposed algorithm with other algorithms for tree construction and maintenance. For tree construction, our *Efficient Wireless Data Center Multicast Tree (EWDCMT)* approach is compared with two algorithms. The first algorithm, denoted as *steiner-tree*, was designed for wired data center networks; the algorithm obtains an optimal multicast tree for each multicast group regardless of the link capacity constraint of each wired link. In order to have a fair comparison, we relax the constraint for *steiner-tree*. Note that relaxing the constraint is beneficial for the performance of *steiner-tree*. The second algorithm, represented as *shortest-path-tree*, was designed as a baseline. The algorithm builds shortest-path trees with the consideration of wired and wireless links in wireless data centers. For each shortest path tree, the algorithm uses wireless links first. Until the available capacity of an access point is exhausted, the algorithm adopts wired links instead. The performance metric is the total amount of transmitted data traffic for all multicast groups.

For tree maintenance, *EWTM-J* and *EWTM-L* were proposed to deal with the cases for node joining and leaving a multicast group. We adopt three algorithms for the performance comparison. *EWDCMT* is considered as the lower bound for the tree maintenance problem. A random approach, denoted by *Random*, randomly chooses wired or wireless links to modify an original multicast tree when receiver joins the multicast group. The third algorithm, represented as *Retrieval*, revokes the resource of a wireless link when the transmission range of the wireless link does not cover any destination and the leaving node is a leaf node. In this experiment, the numbers of multicast groups are 50 and 250, where the size for each group is initially generated by the power-law and the uniform distributions. Then, the number of joining or leaving nodes varies from 100 to 1000, and each node is randomly and subsequently added/removed into/from one of the groups. The performance metric used for tree maintenance is the amount of increased/decreased multicast traffic when the receivers join/leave multicast groups. Finally, we have compared the three algorithms in terms of the execution time when 500 nodes join/leave multicast groups. The experiment is conducted by a desktop computer with Intel CPU I7-3770 3.4GHz and 16GB RAM.

The simulation parameters are listed in Table 2. We measure the simulation results from averaging the results of 500 independent simulations.

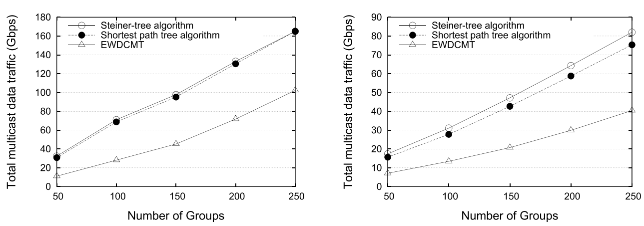
TABLE 2. Parameter settings.

Parameter	Value
Number of top-of-racks	160
Transmission range of a 60GHz wireless access point	10 meters
Available link capacity without background traffic	1 Gbps
Available link capacity with light background traffic	700-1000 Mbps
Available link capacity with heavy background traffic	300-1000 Mbps
Number of multicast groups	50-250
Data rate of each multicast group	1-100000 Kbps
Number of joining/leaving nodes	100-1000

B. SIMULATION RESULTS

1) MULTICAST TREE CONSTRUCTION

Fig. 6 shows the impacts of the number of multicast groups under different group size distributions on the total multicast data traffic. As shown in the figure, the total multicast data traffic increases when the number of multicast groups increases for the three algorithms. The figures intuitively show that more multicast groups increase more multicast data traffic and use more network resources. However, our proposed algorithm can efficiently reduce the total multicast data traffic against *steiner tree* and *shortest path tree*. Comparing Fig. 6(a) with Fig. 6(b), the performance of *shortest path tree* is close to that of *steiner tree* when we consider the uniform group size distribution. The reason is that each multicast group with the uniform group size has a relatively large number of members (destinations). Each member is randomly placed in the wireless data center, so that *shortest path tree* may rapidly exhaust the capacity of each wireless link. Thus, wired links are used instead and the performance of *shortest path tree* is similar to that of *steiner tree*. In contrast, *EWDCMT* significantly reduces more data redundancy, compared with *steiner tree* and *shortest path tree*,



(a) (b)

FIGURE 6. Impacts of the number of multicast groups under (a) the uniform group size distribution and (b) the power-law group size distribution on the total multicast data traffic.

under the uniform group size distribution than under the power-law group size distribution. This is because our algorithm efficiently uses each wireless link and finds each access point that transmits data to as more destinations as possible. When each multicast group has more destinations, our algorithm efficiently utilizes the broadcast advantage of wireless medium for multicast transmissions and evidently reduces the data redundancy of multicast traffic. The simulation results show that *EWDCMT* reduces the total data traffic, compared with *steiner-tree* and *shortest path tree*, from 39% to 66% under the uniform group size shown in Fig. 6(a) and from 48% to 55% under the power-law group size distribution.

Fig. 7 shows the impacts of different background traffic levels on the total multicast data traffic. As we can see in this figure, the total multicast data traffic is higher, when the background traffic load is higher, under *shortest path tree* and *EWDCMT*. The reason is that when the background traffic increases, those efficient wireless links for each multicast group may not afford to satisfy the increased traffic demand. In order to avoid over-utilization, the two algorithms must use other inefficient wireless/wired links for building multicast trees such that data redundancy can be increased. This also explains why the performance of *EWDCMT* is close to those of *shortest path tree* and *steiner-tree* when the background traffic is heavy. On the other hand, the background traffic level does not have any impact for *steiner-tree*, since *steiner-tree* does not consider the link capacity constraint of wired links. Comparing Fig. 7(a) with Fig. 7(b), the result is similar to that in Fig. 6. The performance of our proposed algorithm, compared with *steiner-tree* and *shortest path tree*, is more efficient for reducing total multicast data traffic under the uniform group size distribution, shown in 7(a), than under the power-law group size distribution, shown in 7(b). The simulation results show that *EWDCMT* outperforms *steiner-tree* and *shortest path tree*. The reduction is about 56% under the uniform group size distribution and is about 52% under the power-law group size distribution.

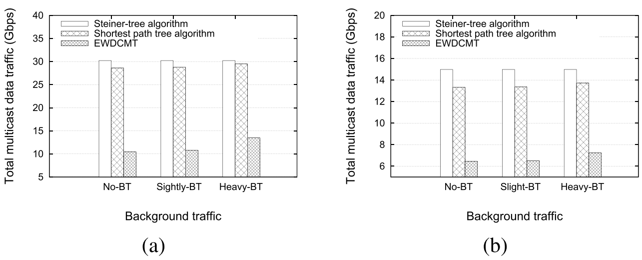


FIGURE 7. Impacts of the number of multicast groups for (a) the uniform group size distribution and (b) the power-law group size distribution on the total multicast data traffic under 50 multicast groups.

In addition to the topology used by Microsoft and the synthetic input of data rates for multicast traffic, we collected real traces of MepReduce in Chunghwa Telecom data center to evaluate the performance of *EWDCMT*. In this data center, there are six top-of-racks and 120 servers as a cluster for cooperating computation, and the six top-of-racks are arranged in a straight line. Based on the real traces, the corresponding data rates can be parsed. Fig. 8 shows the impact of the number of multicast groups on the total multicast data traffic based on the real traces. The result is consistent with the results following the settings by Microsoft. In this figure, we found that our proposed algorithm can save at most 86% of the total multicast data traffic in comparison with *steiner-tree* and *shortest-path-tree*, which indicates that our proposed algorithm efficiently uses network bandwidth for multicast transmissions to reduce unnecessary multicast traffic in a realistic environment.

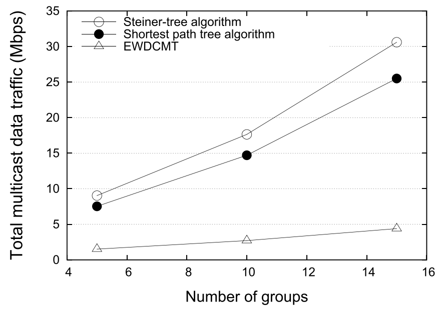


FIGURE 8. Impact of the number of multicast groups on the total multicast data traffic by the real traces of MapReduce from Chunghwa Telecom.

2) MULTICAST TREE MAINTENANCE

Fig. 9 shows the impacts of the number of joining nodes with power-law distribution on the amount of the increased multicast traffic when there are 50 and 250 multicast groups. We observe that the amount of the multicast traffic increases as the number of joining nodes increases for *Random*, *EWDCMT* and *EWTM-J*. This result can be expected because more joining nodes imply more traffic requests. Compared with *Random*, our proposed algorithm *EWTM-J* can save more unnecessary multicast traffic, because *EWTM-J* can efficiently maintain the used wireless links or can find substitute paths for the victims groups. Moreover, the performance of our algorithm is close to that of *EWDCMT*. By comparing Fig. 9(a) with Fig. 9(b), the performance of *EWDCMT* and

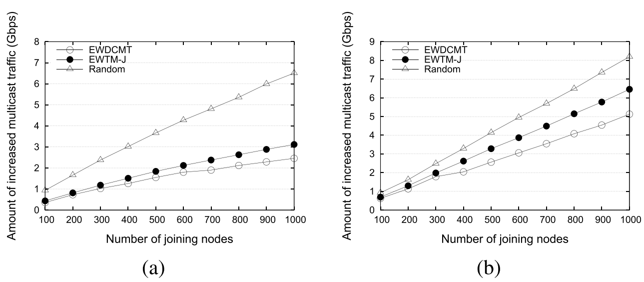
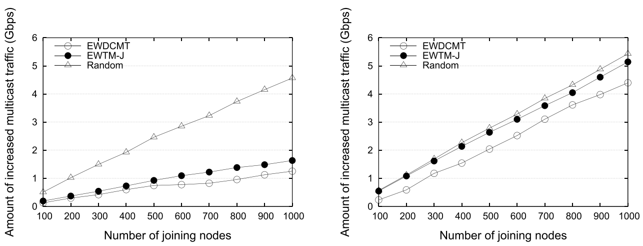


FIGURE 9. Impacts of the number of joining nodes with the power-law group size distribution on the amount of the increased multicast traffic under (a) 50 multicast groups and (b) 250 multicast groups.

EWTM-J is closer to *Random* under 250 groups than under 50 groups. This is because when the group size increases, more groups have to compete wireless resources and be the victim groups to use wired links. The simulation results show that compared with *Random*, *EWTM-J* can reduce the amount of the increased multicast traffic to 53% under the case of 50 multicast groups. Moreover, *EWTM-J* generates the amount of the multicast traffic at most 26% more than *EWDCMT*.

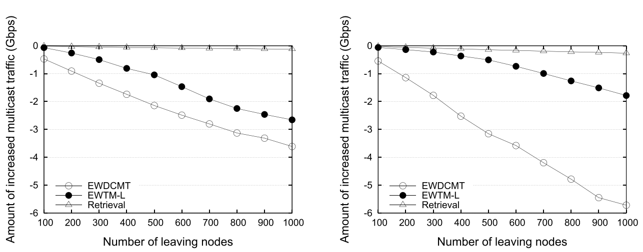


(a) (b)

FIGURE 10. Impacts of the number of joining nodes with uniform group size distribution on the amount of the increased multicast traffic under (a) 50 multicast groups and (b) 250 multicast groups.

Fig. 10 shows the impacts of the number of joining nodes with uniform distribution on the amount of the increased multicast traffic for 50 and 250 multicast groups. As shown in Fig. 10(a), the result is similar to that in Fig. 9(a) when the group size is 50. On the other hand, when the group size is 250, the performance of the three algorithms is similar as shown in Fig. 10(b). This phenomenon is due to that the three algorithms will exhaust wireless resources under 250 groups with uniform distribution and wired links are unavoidably used.

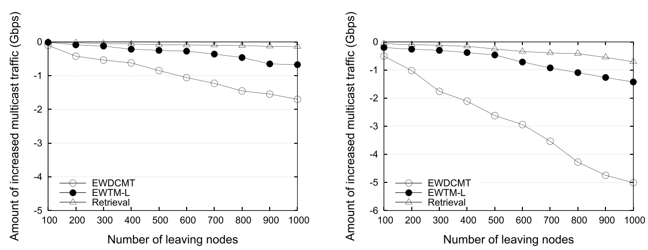
Fig. 11 shows the impacts of the number of leaving nodes with power-law distribution on the amount of the increased multicast traffic under 50 and 250 multicast groups. The amount of the multicast traffic decreases when the number of leaving nodes increases for all of the three algorithms. The reason is that more wired and wireless resources are released for optimizing the resource allocation for the remaining nodes when there are more leaving nodes. As shown in Fig. 11(a), the decrease on the amount of the multicast traffic is more evident under *EWTM-L* than under *Retrieval*. This is because our algorithm tries to revoke all of the unused links in the transmission path for a group and reallocate the resources to other groups, while *Retrieval* only considers the wireless link used by the leaving nodes. Comparing Fig. 11(a) with Fig. 11(b), *EWTM-L* can release more resources under 50 groups than under 250 groups. This phenomenon is that when the number of groups is fewer, the leaving nodes are very likely to belong to the same group such that more resource can be released and reallocated to other groups. On the other hand, when there are more number of groups, the leaving nodes are probably distributed to different groups such that the wireless resources for the leaving nodes cannot be completely released.



(a) (b)

FIGURE 11. Impacts of the number of leaving nodes with power-law group size distribution on the amount of the increased multicast traffic under (a) 50 multicast groups and (b) 250 multicast groups.

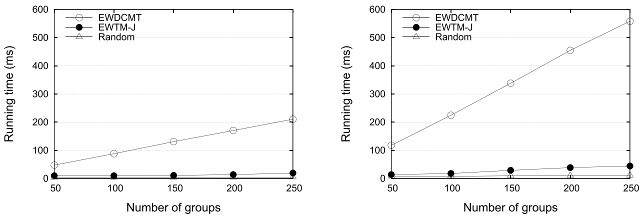
Fig. 12 shows the impacts of the number of leaving nodes with uniform distribution on the amount of the increased multicast traffic under 50 and 250 multicast groups. As shown in Fig. 12(a) and 12(b), under the uniform distribution, the decreased multicast traffic is not evident for *EWTM-L*. The reason is similar to that for Fig. 11(b). *EWTM-L* only reallocates the released wireless resource. When the resource is occupied by few nodes, our proposed algorithm does not have a chance to reallocate the wireless resources. In contrast, *EWDCMT* can reallocate all the wired and wireless resource for all groups. When there are more leaving nodes, *EWDCMT* can release more bandwidth as expected. Compared with Fig. 12(a) and Fig. 12(b), *EWDCMT* can reduce more data traffic under 50 groups than under 250 groups. This is because when there are more nodes in a group, *EWDCMT* can more efficiently reallocate wireless transmissions to reduce data redundancy.



(a) (b)

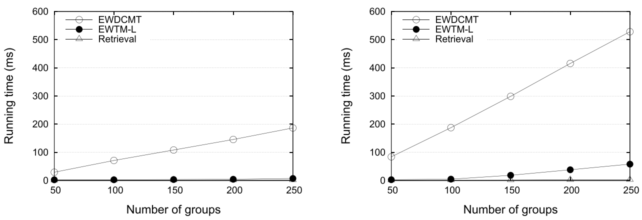
FIGURE 12. Impacts of the number of leaving nodes with the uniform group size distribution on the amount of the increased multicast traffic under (a) 50 multicast groups and (b) 250 multicast groups.

Figs. 13 and 14 respectively show the impacts of the number of groups on the average running time required for each algorithm. From these figures, we observe that the running time significantly increases when the number of groups increases for *EWDCMT*. In contrast, the increase of the running time is not so significant with the number of groups for *EWTM-J*, *Random*, *EWTM-L*, and *Retrieval*. The reason is that *EWDCMT* has to rebuild whole multicast trees for minimizing data redundancy, while the other four algorithms only reconstruct part of the multicast trees to



(a) (b)

FIGURE 13. Impacts of the number of groups with 500 joining nodes on the average running time required for each algorithm under (a) power-law group size distribution and (b) uniform group size distribution.



(a) (b)

FIGURE 14. Impacts of the number of groups with 500 leaving nodes on the average running time required for each algorithm under (a) power-law group size distribution and (b) uniform group size distribution.

satisfy the requests. We also find that the running time under uniform group size distribution is higher than that under power-law group size distribution. This is because by following power-law distribution, most of the groups tend to be small and then each algorithm spends less time to handle tree maintenance due to node joining and leaving. The simulation results show that the average running time required by our proposed algorithms (i.e., *EWTM-J* and *EWTM-L*) are shorter than 600μs for processing a multicast request (flow). Recent researches indicate that in data centers, 70% of the flows are delay-sensitive short flows [26], which require their flow setup time shorter than 1ms [27]. The result confirms that the our proposed tree maintenance algorithms are applicable to data centers.

By comparing the algorithm for node joining (i.e., *EWTM-J*) and leaving (i.e., *EWTM-L*), we can observe an interesting phenomenon. *EWTM-J* can reduce more data redundancy when the scale of the system is large. In contrast, the efficacy of *EWTM-J* is more evident when the number of nodes is fewer (e.g., when the group size is 50 with power-law distribution). Moreover, the performance of *EWTM-J* and *EWTM-L* degrades as the number of the joining/leaving nodes increases. A open issue is when we should adopt *EWDCMT* to rebuild the whole multicast trees of all groups to gain a better output. In fact, there exists a trade-off between the system performance and the computational complexity. System operators can design their policies according to their system performance requirements. We do not focus on the issue in this paper, and it can be one of the future directions for extending the research.

VII. CONCLUSION

In this paper, we have addressed the group communication issue raised in wireless data center networks. We explored the multicast tree construction and maintenance problems with the coexistence of wired and wireless links. The objective of this paper is to minimize the total multicast traffic. We proved $\mathcal{NP}$-hardness of the target problems. For the tree construction problem, we proposed a heuristic algorithm to efficiently use wireless transmission links. For the tree maintenance problem, a low-complexity solution was developed to adjust the multicast trees when their receivers join/leave. Finally, we conducted a series of simulations to evaluate the performance of our proposed algorithms. The simulation results demonstrated that our proposed algorithms are effective for reducing the total multicast traffic. We also observed some useful insights which can be used to the design of multicast tree construction and maintenance for wireless data center networks.

REFERENCES

[1] S. Ghemawat, H. Gobioff, and S.-T. Leungm, "The Google file system," in *Proc. 19th ACM SOSP*, 2003, pp. 29–43.
[2] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proc. 6th Conf. Symp. OSDI*, 2004, p. 10.
[3] K. Nagaraj, H. Khandelwal, C. Killian, and R. R. Kompella, "Hierarchy-aware distributed overlays in data centers using DC2," in *Proc. 4th Int. Conf. COMSNETS*, Jan. 2012, pp. 1–10.
[4] J. Cao *et al.*, "Datacast: A scalable and efficient reliable group data delivery service for data centers," in *Proc. ACM 8th Int. Conf. CoNEXT*, 2012, pp. 37–48.
[5] S. Kandula, J. Padhye, and P. Bahl, "Flyways to de-congest data center networks," in *Proc. ACM Workshop Hot Topics Netw.*, 2009, pp. 1–6.
[6] D. Halperin, S. Kandula, J. Padhye, P. Bahl, and D. Wetherall, "Augmenting data center networks with multi-gigabit wireless links," in *Proc. ACM SIGCOMM Conf.*, 2011, pp. 38–49.
[7] X. Zhou *et al.*, "Mirror mirror on the ceiling: Flexible wireless links for data centers," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2012, pp. 443–454.
[8] Y. Katayama, K. Takano, Y. Kohda, N. Ohba, and D. Nakano, "Wireless data center networking with steered-beam mmWave links," in *Proc. IEEE WCNC*, Mar. 2011, pp. 2179–2184.
[9] J.-Y. Shin, E. G. Sirer, H. Weatherspoon, and D. Kirovski, "On the feasibility of completely wireless datacenters," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1666–1679, Oct. 2013.
[10] S. Deering, *Host Extensions for IP Multicasting*, document RFC 1112, 1989.
[11] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, *Internet Group Management Protocol*, document RFC 3376, 2002.
[12] Y. Yang, J. Wang, and M. Yang, "A service-centric multicast architecture and routing protocol," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 1, pp. 35–51, Jan. 2008.
[13] D. Li, J. Yu, J. Yu, and J. Wu, "Exploring efficient and scalable multicast routing in future data center networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1368–1376.
[14] L. Junhai, Y. Danxia, X. Liu, and F. Mingyu, "A survey of multicast routing protocols for mobile ad-hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 78–91, First Quarter 2009.
[15] J. J. Garcia-Luna-Aceves and E. L. Madruga, "The core-assisted mesh protocol," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1380–1394, Aug. 1999.
[16] K. Chen and K. Nahrstedt, "Effective location-guided tree construction algorithms for small group multicast in MANET," in *Proc. 21st Annu. Joint Conf. IEEE INFOCOM*, Jun. 2002, pp. 1180–1189.
[17] J. Biswas, M. Barai, and S. K. Nandy, "Efficient hybrid multicast routing protocol for ad-hoc wireless networks," in *Proc. 29th Annu. IEEE Int. Conf. LCN*, Nov. 2004, pp. 180–187.

[18] Y. Vigfusson *et al.*, "Dr. multicast: *Rx* for data center communication scalability," in *Proc. ACM 5th EuroSys*, 2010, pp. 349–362.
[19] D. Li, Y. Li, J. Wu, S. Su, and J. Yu, "ESM: Efficient and scalable data center multicast routing," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 944–955, Jun. 2012.
[20] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2008, pp. 63–74.
[21] C. Guo *et al.*, "BCube: A high performance, server-centric network architecture for modular data centers," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 63–74.
[22] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.
[23] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1st ed. New York, NY, USA: Freeman, Jan. 1979.
[24] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," in *Proc. 1st ACM SIGCOMM Workshop Res. Enterprise Netw.*, 2010, pp. 65–72.
[25] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: Measurements & analysis," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas.*, 2009, pp. 202–208.
[26] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, 2010, pp. 267–280.
[27] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: Scaling flow management for high-performance networks," in *Proc. ACM SIGCOMM Conf.*, 2011, pp. 254–265.

CHING-CHIH CHUANG (S'13) received the B.S. degree in computer science and information engineering from I-Shou University, in 2008, and the M.S. degree in computer science and information engineering from National Chung Cheng University, in 2010. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, National Taiwan University. His research interests include data center networks and software defined networking.

YA-JU YU received the B.S. degree in computer and communication engineering from the National Kaohsiung First University of Science and Technology, in 2005, the M.S. degree in communication engineering from National Central University, in 2007, and the Ph.D. degree from the Graduate Institute of Networking and Multimedia, National Taiwan University, in 2012. He is currently a Senior Engineer with the Smart Network System Institute, Institute for Information Industry, Taiwan. His research interests include wireless mobile networks, multimedia communications, and cloud datacenter networking.

AI-CHUN PANG (SM'95) received the B.S., M.S., and Ph.D. degrees in computer science and information engineering from National Chiao Tung University, Taiwan, in 1996, 1998, and 2002, respectively. She joined the Department of Computer Science and Information Engineering, National Taiwan University (NTU), Taipei, Taiwan, in 2002. She is currently the Director of the Graduate Institute of Networking and Multimedia (INM), NTU, and a Professor with the Department of Computer Science and Information Engineering and INM. She is also an Adjunct Research Fellow with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan. She has co-authored a book entitled *Wireless and Mobile All-IP Networks* (John Wiley Sons Inc.). Her research interests include the design and analysis of wireless and multimedia networking, mobile communications, and cloud data center networking. She was a recipient of the Outstanding Teaching Award at NTU in 2010, the Investigative Research Award of the Pan Wen Yuan Foundation in 2006, the Wu Ta You Memorial Award of the National Science Council in 2007, the Excellent Young Engineer Award of the Chinese Institute of Electrical Engineering in 2007, and the K. T. Li Award for Young Researchers of the ACM Taipei/Taiwan Chapter in 2007. She was also a recipient of the Republic of China Distinguished Women Medal in 2009. She was a Guest Editor of the *IEEE Wireless Communications*, and is an Associate Editor of *Wireless Networks* and *Security and Communication Networks*. She served on the Technical Program Committee of many international conferences, including the IEEE INFOCOM, the IEEE GLOBE-COM, the IEEE ICC, and the IEEE VTC.

HSUEH-WEN TSENG (M'11) received the Ph.D. degrees in computer science and information engineering from National Taiwan University, in 2009. He is currently an Assistant Professor of Computer Science and Engineering with National Chung Hsing University. His research interests include cloud computing and networking, networks-on-chip, design, analysis, and implementation of network protocols, and wireless networks.

HSIN-PENG LIN received the B.S. degree in transportation management from Tamkang University, in 1996, and the M.S. degree in transportation and communication management from National Cheng Kung University, in 1998. He is currently a Researcher with Chunghwa Telecom Laboratories and a part-time Ph.D. Student with the Graduate Institute of Computer Science and Information Engineering, National Taiwan University. His research interests include multimedia communications, cloud data center networking, and wearable devices.