WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
*Wirel. Commun. Mob. Comput.* (2008)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.629

# Fair inter-TAP routing and backhaul assignment for wireless mesh networks

Frank Y. S. Lin and Y. F. Wen*,†
*Department of Information Management, National Taiwan University, Taiwan, R.O.C.*

## Summary

Although limiting the number of backhauls, specifically chosen transit access points (TAPs) that forward traffic from other TAPs, reduces the overall costs of a wireless mesh network (WMN), an egress bottleneck is induced, which aggregates traffic and limits the bandwidth. To avoid such problems while working to minimize budgetary expenses, we balanced traffic flow on 'to-be-determined' backhauls and adjacent links, a mixed nonlinear- and integer-programming problem that minimizes the aggregated flow subject to budget, backhaul assignment, top-level load-balanced routing, and link capacity constraints. Two algorithms are proposed, weighted backhaul assignment (WBA) and greedy load-balanced routing (GLBR), that operate in conjunction with Lagrangean relaxation (LR), used for constructing LR-based heuristics and also as a means of quantification and evaluation of the proposed algorithms. Experiment results show that the proposed algorithms achieve near-optimization, outperforming related solutions. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: cluster; fairness; load-balancing; mesh networks; routing; optimization

## 1. Introduction

The slogan, 'Anywhere, anytime,' has become entrenched in the wireless Internet literature, yet ubiquitous access comes with costs, a constant concern of consumers and providers alike. 3G and wireless local area network (WLAN) services are an inexpensive means of providing last-mile connectivity to the Internet [1], facilitating 24-hour coverage public wireless local area networks (PWLAN) [1,2].

By employing a mesh network on the last hop, overall Internet service provider (ISP) costs of deployment and maintenance may be reduced [3]. The typical wireless mesh network (WMN), shown in Figure 1, is comprised of transit access point (TAPs) [4] associated with wireless links that provide sufficiently wide coverage for infrastructures such as those at wireless hot spots. The backhauls (i.e., egress interfaces and links) provide the data services that enable mobile devices (MDs) to access the Internet through either wired or wireless interfaces. It includes the interface and outgoing link as egress wired (e.g., fiber and T1) or wireless (e.g., WiMAX) links. For further on mesh networks, refer to Reference [5]. The determination of routing is less complex for TAPs that are located in fixed positions such as those mounted on traffic lights.

The main motivation of this paper is the reduction of the negative impacts that arise from backhaul and routing assignments:

*Correspondence to: Y. F. Wen, Department of Information Management, Chinese Culture University, 55, Hwa-Kang Road, Yang-Ming Shan, Taipei 111, Taiwan, R.O.C.
†E-mail: yeanfu@gmail.com

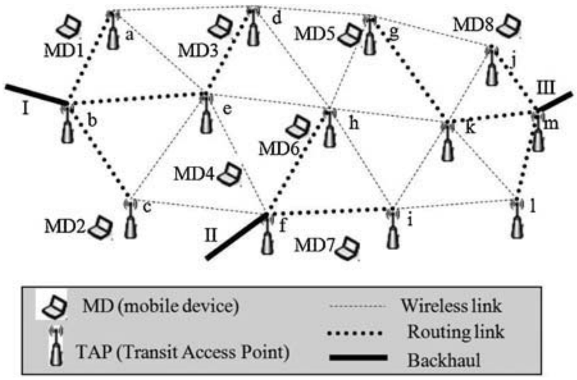


Fig. 1. A mesh network constructed with a BS-oriented structure connecting MDs to a TAP and an *ad hoc* structure connecting TAPs to a wired network with backhauls. Wireless links enable this type of network to cover a large area.

(a) **Backhaul assignment.** A primary concern of our work is network cost. Therefore, we allow restrictions on the number of backhauls that may be installed. Additionally, once a TAP has been selected to be attached with a backhaul, it remains so. With such limited availability of backhauls, a resulting network egress bottleneck hinders smooth flow.

When the suitable backhaul assigned could let traffic distribute well, the amount of flow travel on the network is less. Thus, we try to find the centric of the high traffic area. Then, it could reduce the high load traffic transmit in long path and divide the traffic load balancing among the backhauls. These help to balance the traffic load to fulfill fairness issue.

The assignment of backhauls bears significant similarity to a typical cluster-head (CH) assignment. Many CH assignment methods focus on balancing cluster sizes, for example, max-min *d*-hop cluster [6], linked cluster algorithm (LCA) [7], and highest degree (HD) algorithm [8]. CH assignment was proven to be NP-complete in References [6] and [9]. Other researchers [10–12] have focused on fair selection of CHs.

Significant performance improvements have been achieved by such an assignment method. In Reference [11], clustering provided a means by which CHs made use of link state information. The gravitational cluster routing (GCR) protocol [13] increases stability when providing coverage to dense areas by choosing CHs according to 'local maximum degree.' We employ a similar concept for our proposed algorithm, thereby reducing the average transmission cost.

(b) **Load-balanced inter-TAP routing.** For primarily budgetary reasons, multiple-hops from backhauls may be required in our networks. As routing algorithms, including *ad hoc* shortest path routing algorithms [14–17] and hierarchical and hybrid routing protocols [18–20], minimize path lengths without taking load capabilities into consideration, traffic discrepancies may occur, resulting in unequal loads and long delays for some TAPs. Thus, the load balancing of routing in orthogonal wireless network is addressed in this paper.

When the relations between fairness, network capacity, and congestion controls are considered, fairness means load balanced that distributes the traffic among the nodes with backhauls and their branches. It lets the amount of traffic much lower than the capacity and reduces the congestion problem. In other words, fairness is an efficient congestion control approach. The congestion problem is also affected by the network capacity. If the traffic load is close to capacity, the traffic jam occurs, causes congestion, and decreases network performance. Thus, if we could distribute the traffic among the network to balance the traffic below the capacity, the congestion situation is reduced.

For the hybrid network treats the different transmission lines. The backhaul interface could be wired or wireless. The wired links are more stable than wireless networks and they can take more traffic with broadband wired lines. However, the cost is high. In addition, the node attached with a backhaul as a tree root still needs to process all branches subtree traffic and comprises a bottleneck.

In References [21] and [22], the authors have balanced the traffic load on the incoming link of the egress node that installed a backhaul, considered to be the primary bottleneck, and have focused on determining a top-level load-balanced tree (TLBT) topology, as shown in Figure 2(a). A TLBT topology applies to networks with multiple sources that route to a single destination.

A WMN is a forest if many egresses, or CHs, exist in the network. For this paper, the concept of a TLBT was extended to a top-level load-balanced forest (TLBF), an example of which is shown in Figure 2(b). A TLBF is load balanced such that the given fair amount of traffic is carried by each backhaul and its branches, the links that are closest to the backhauls.



DOI: 10.1002/wcm

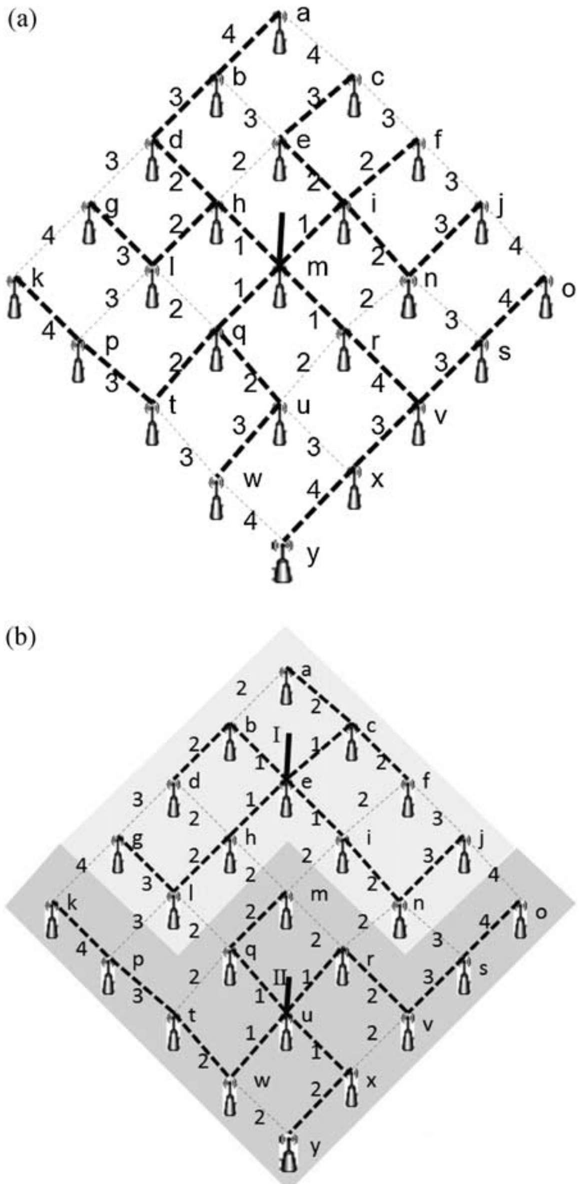


Fig. 2. Graphs for TLBT and TLBF topologies, for which one (TLBT) or more (TLBF) backhauls are assigned (the symbols and their meaning are shown in Figure 1). (a) TLBT (backhaul I on node *m*), (b) TLBF (backhaul I on node *e* and II on node *u*).

Definition:

- TLBT means to find the load balanced the traffic load on each branch among the root of a tree.
- TLBF means to find both load-balanced traffic load on each branch of a TAP (which is installed a backhaul) and each backhaul.

The primary motivation of this paper is the reduction of the negative impacts that arise from the two issues above. Our proposed algorithms, WBA, GLBR, and the LR-based heuristic operate by determining a minimal aggregated flow for each link and by evenly distributing loads over the 'to-be-determined' backhauls. In computational experiments, the LR-based approach, which has been used to solve many famous NP-complete problems [23], was used to improve the timing and quality requirements, and furthermore, was used to determine near-optimal decisions. The LR approach provides a means of computing a lower bound (LB), used to determine the minimum gap between the value returned by our algorithms and the optimal solution.

The remainder of this paper is organized as follows. In Section 2, we provide a mathematical formulation of TLBF routing, including the backhaul assignment problem. Then, in Section 3, we propose algorithms for backhaul assignment and TLBF routing. In Section 4, we illustrate LR-based solutions. Experimental results show near-optimization as detailed in Section 5, where our approach is compared with existing methods. Finally, in Section 6, we present conclusions.

## 2. Problem Description and Formulation

On a network modeled as a graph $G(V, L)$, where $V$ vertices represent TAPs distributed on an area with direct links $L$ between TAPs within the transmission range, a TLBF topology partitions the graph $G$ into several connected sub-graphs or cliques based on the aggregated flow of each link. We assume as given that MD-to-TAP and TAP-to-TAP transmissions occur on orthogonal channels, such that: (a) each TAP has a directional antenna; (b) for each TAP, the arrival of new traffic is modeled as a Poisson process with rate $\gamma_s$ (units/s, where the 'units' could be packets, Mega-bits, or other measures); and (c) all flows are transmitted via 'to-be-determined' backhauls. Accordingly, the problem description is summarized as follows:

**Given:**

- The set of TAPs $V$.
- The set of candidate backhauls $B$.
- The set of candidate paths for a TAP to reach a backhaul $P_{sb}$.
- The set of links $L$.
- The link capacity $C_{(u,v)}$, where $(u, v) \in L$.
- The traffic requirement for each TAP with rate $\gamma_s$ (units/s).



DOI: 10.1002/wcm

- The maximum number of hops, $H_s$, along the shortest path from source node to reach the most distant destination. (Used for calculating the aggregated flow.)

**Object:**

To minimize the sum of aggregated flows of selected links.

**Subject to the following constraints:**

- Backhaul assignment: the total cost of assigned backhauls is limited within the budget $\varphi$.
- Backhaul selection: once backhauls have been assigned, each TAP must select one of the backhauls as a gateway.
- Routing: one path reach to a backhaul must be found in order to transmit/receive data to/from.
- Capacity: the aggregated flow of each link is limited by capacity constraint.
- Fairness index (FI): the amount of aggregated flow on each backhaul and their adjacent links must satisfy the given FI value.

**To determine:**

- Which TAP is selected to attach with a backhaul.
- A backhaul is selected to transmit data to for each TAP.
- Routing path from a TAP to a backhaul.
- Whether a link should be selected for the routing path.
- A top-level load-balanced forest (TLBF).
- Aggregated flow on each selected link.
- Aggregated flow on each backhaul.

The problem is represented as the following formulations. The objective function (1) is the minimum of the aggregated flow for all connection links:

$$Z_{\mathrm{IP}} = \min \sum_{(u,v)\in L} f_{(u,v)} \tag{1}$$

subject to the following constraints:

(a) **Backhaul selection constraints.** Decision variable $z_{sb} \in \{0, 1\}$ is used to indicate if candidate backhaul $b$ has been selected as the gateway for node $s$. Each TAP routes to only one backhaul. The total number of backhauls assigned to each TAP is 1, as in Constraint (2).

$$\sum_{b\in B} z_{sb} = 1, \quad \forall s \in V \tag{2}$$

(b) **Backhaul assignment constraints.** A 0-1 decision variable $\eta_b$ is used to determine when a backhaul $b$ is chosen as a backhaul. Constraint (3) stipulates that if at least one TAP $s$ chooses $b$ as a backhaul, $\eta_b$ must be set to 1. However, if no TAP chooses $b$ as a backhaul, $\eta_b$ must be set to 0 in order to reduce costs for other assignment requirements. The total cost is subject to the budget Constraint (4). The summation of all built costs $\phi_b$ of assigned backhauls, denoted by $\eta_b = 1$, is not to exceed the budget $\varphi$.

$$z_{sb} \le \eta_b, \quad \forall b \in B,\ s \in V \tag{3}$$

$$\sum_{b\in B} \phi_b \eta_b \le \varphi \tag{4}$$

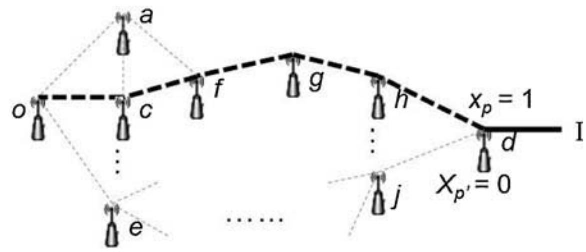


Fig. 3. All paths from $s$ (i.e., TAP $o$) to $b$ (i.e., backhaul I) are included in the set $P_{sb}$. The path above, shown by the bold dash line $\{o, c, f, g, h, d\}$, is the only active path in this set, so for that path $x_p = 1$. As there are no other active paths, for any other path $p'$ (e.g., path $\{o, e, \ldots, j, d\}$), $x_{p'} = 0$ (the symbols and their meaning are shown in Figure 1).

(c) **Path constraints.** When decision variable $x_p = 1$, it indicates that the path $p \in P_{sb}$ is the path used to connect the TAP to the backhaul; $x_p = 0$ means that it is not used. Figure 3 shows an example of the decision variable. Each TAP can exist on only one path, shown as

$$\sum_{b\in B} \sum_{p\in P_{sb}} x_p = 1, \quad \forall s \in V \tag{5}$$

The variables $x_p$ and $z_{sb}$ are related such that once a path $p$ has been selected, the variable $z_{sb}$ must be set to 1. This constraint is described by Equation (6), where the limit on the right-hand side of Equation (5) means that only one path from the set of candidate paths $P_{sb}$ is selected.

$$\sum_{b\in B} z_{sb} \le \sum_{b\in B} \sum_{p\in P_{sb}} x_p, \quad \forall s \in V \tag{6}$$

Once node $s$ has selected backhaul $b$ as its gateway (i.e., $z_{sb} = 1$), one path is selected from the set of $P_{sb}$ to connect TAP $s$ to backhaul $b$, as shown in



DOI: 10.1002/wcm

Figure 3. Once the path has been determined and $x_p$ has been set to 1, all other paths $p'$ are set to 0 (i.e., $x_{p'} = 0$). Note that we are interested in determining only one path, and that path can be found by the proposed algorithm. Thus, the set of all candidate paths neither needs to be determined nor initially listed.

(d) **Link constraints.** The decision variable $y_{(u,v)} = 1$ indicates that a link $(u, v) \in L$ is used; $y_{(u,v)} = 0$ indicates that a link $(u, v) \in L$ is not used. The relationship between path $x_p$ and link $y_{(u,v)}$ is such that once a path $p$ is selected and a link $(u, v)$ is on the path, the decision variable $y_{(u,v)}$ must be set to 1. For example, in Figure 3, path $p$ is selected. The decision variable $y_{(u,v)}$ is set to 1 for the links $(o, c)$, $(c, f)$, $(f, g)$, $(g, h)$, and $(h, d)$; for all other links, to 0. This constraint is described by

$$\sum_{p \in P_{sb}} x_p \delta_{p(u,v)} \le y_{(u,v)}, \quad \forall s \in V;\ b \in B;\ (u, v) \in L \tag{7}$$

where $\delta_{p(u,v)}$ is an indicator function, indicating that link $(u, v)$ is on the path $p$, where $p \in P_{sb}$.

Since the problem is to find a forest constructed by many trees, a link constraint is needed to describe the forest structure, constraining that the out-degree links from a TAP is not larger than 1. However, for all TAPs assigned to be backhauls, the number of out-degree links is equal to 0, as the wired-line is not considered an out-degree link for the purposes of our model. These situations are described as:

$$\sum_{v \in V} y_{(u,v)} \le 1, \quad \forall u \in V \tag{8}$$

$$\sum_{v \in V} y_{(b,v)} = 0 \tag{9}$$

According to the previous description, the total number of selected links will be equal to the total number of TAPs minus the total number of given backhauls. (In the point view by graph theorem, the construct of TLBF is forest and then the number of total links is the number of nodes minus the number of roots.) Thus, Constraint (10) is added to limit the number of selected links.

$$\sum_{(u,v) \in L} y_{(u,v)} = |V| - |B| \tag{10}$$

where $|V|$ is the number of nodes in set $V$ and $|B|$ is the number of backhauls in set $B$.

The number of hops on a selected path $p$ is limited to be less than a given value $H_s$ for any node $s$. On the left-hand side of Equation (11), each node s selects only one backhaul $b$ as its gateway, so only one path $p$ is selected. Accordingly, if a link $(u, v)$ is on a path $p$, then $x_p \delta_{p(u,v)}$ is equal to 1. Thus, the number of hops on the path $p$ is calculated on the left-hand side (e.g., five links satisfy $x_p \delta_{p(u,v)} = 1$, so in Figure 3, there are five hops for path $p$). This constraint is used for backhaul assignment.

$$\sum_{b \in B} \sum_{p \in P_{sb}} \sum_{(u,v) \in L} x_p \delta_{p(u,v)} \le H_s, \quad \forall s \in V \tag{11}$$

(e) **Flow constraints.** Once the backhaul and routing assignments have been determined, a variable amount of traffic $\gamma_s$ is randomly generated for each source node $s$. Thus, for each link, the aggregated flow $f_{(u,v)}$ is calculated as Equation (12) for all selected paths $p$ via link $(u, v)$ for node $s$.

$$\sum_{s \in V} \sum_{b \in B} \sum_{p \in P_{sb}} x_p \delta_{p(u,v)} \gamma_s \le f_{(u,v)}, \quad \forall (u, v) \in L \tag{12}$$

So that it does not exceed the given link capacity, the aggregated flow is limited by $C_{(u,v)}$; this capacity constraint is formulated as

$$0 \le f_{(u,v)} \le C_{(u,v)}, \quad \forall (u, v) \in L \tag{13}$$

The aggregated flow over each backhaul, denoted by $\beta_b$, is calculated by summing the aggregated flow of all adjacent branches, shown as

$$\sum_{u \in V} f_{(u,b)} \le \beta_b, \quad \forall b \in B;\ (u, b) \in L \tag{14}$$

(f) **FI constraints.** First, we introduce the FI techniques which have the following four properties: (i) population size independence: the index applicable to any number of users, finite or infinite; (ii) scale independence: the index can be independent of scale, that is, the unit of measurement should not matter; (iii) continuity: the index can be continuous so that any slight change in allocation should show up in the FI; and (iv) bounded between 0 and 1: a totally fair system has a fairness of 1, while a totally unfair system approaches 0 [24].

The traffic load balanced among the backhauls and branches of the nodes attached with backhaul interfaces are quantified by FI function. This function comes from the Chebyshev Sum Inequality [25], which follows all branches' aggregated flow $f_{(u,b)}$ associate to the node with backhaul $b$. Then, the inequality would be:

$$\left(\sum_{(u,b)\in L} f_{(u,b)}\right)^2 \leq E_b \sum_{(u,b)\in L} f_{(u,b)}^2 \qquad (15)$$

where $E_b$ is the number of branches associate to a backhaul $b$.

To assess the degree of balance among the different branch's traffic load, the inequalities will become

$$\frac{\left(\sum_{u\in V} f_{(u,b)}\right)^2}{E_b \sum_{u\in V} f_{(u,b)}^2} \leq 1 \qquad (16)$$

Only when all traffic loads of all branches, this value is equal to 1. Otherwise, the value is less than 1 when any branch traffic load is different from others. The degree of unbalance is reflected in this function.

Accordingly, we add a constant value $\alpha'$ to enable traffic load balance. The aggregated flow $f_{(u,v)}$ for each link is then put into the FI equation [24], to achieve the given FI value, denoted by $\alpha'$, as shown in Reference [24]. If the value of $\alpha'$ is set to 1, the loads will be more evenly distributed on each link associated with the backhaul, but the possibility exists that no feasible solution may be found. Thus, $\alpha'$ is set to a reasonable value that can enable us to obtain a feasible solution. This value is set based on the experimental results.

$$\alpha' \leq \frac{\left(\sum_{u\in V} f_{(u,b)}\right)^2}{E_b \sum_{u\in V} f_{(u,b)}^2} \qquad (17)$$

We also calculate the aggregated flow balanced on each backhaul $b$, and put it into the FI equation to achieve the given fairness value $\alpha''$. This constraint is

$$\alpha'' \leq \frac{\left(\sum_{b\in B} \beta_b\right)^2}{|B| \sum_{b\in B} \beta_b^2} \qquad (18)$$

where $|B|$ is the number of backhauls in set $B$.

## 3. The Proposed Algorithms

This section encompasses the details of assigning backhauls and TLBF routing algorithms.

### 3.1. Weighted Backhaul Assignment Algorithm

We propose a weighted backhaul assignment (WBA) algorithm that solves the backhaul assignment problem (i.e., Constraints (3) and (4)), the pseudocode of which is shown as follows. The candidate TAP with the highest weight is selected as a backhaul after adjustments ensure the aggregated flow to be within capacity. With the determination of the average backhaul installation cost $c_a$ (line no. 1), the assignment of approximately $n$ backhauls is expected (line no. 2). The total system flow $t_f$ divided by the expected number of backhauls $n$ gives the expected flow $t_b$ to each candidate backhaul (line no. 3–9). Calculating installation costs $\phi_v$ according to the loading of the neighborhood, and the aggregated flow $\beta_v$ within $H_v$ hops of each TAP, allows us to determine the ratio of the aggregated flow to the unit cost for each candidate backhaul, used as the weight $w_v$.

Finally, as long as some budget remains, TAPs are repeatedly selected as backhauls (line no. 12–25) as follows: (i) the TAP with the highest weight $w_v$ is assigned to be a backhaul (line no. 13). Then, the node $v$ is marked and included in the set $T$. The build-up cost $\phi_v$ is added into total cost $t_c$ (line no. 15 and 16). Before the service node is marked, the aggregated flow $t_a$ is set to be 0; (ii) the smallest flow from the adjacent TAPs is sequentially removed from the dominant set $T$ until the aggregated flow falls below $t_b$ (line no. 18–22). The minimum link flow of the unmarked adjacent TAP $u$ that connects to the minimum aggregate branch flow is found and marked as served node (line no. 19). The aggregated flow of current selected backhaul is calculated (line no. 21); (iii) the weights of unselected backhauls are recalculated. They are recalculated the aggregated flow, $\beta_v$, within the $H_v$ hop adjacent TAPs which is marked as FALSE (line no. 23) and then divided by their build-up cost (line no. 24).

Figure 4 shows an example of the WBA algorithm where the aggregated flow for each TAP is a 2-hop neighborhood. Once all of the backhauls have been assigned, the routing paths of the TLBF are determined by the GLBR algorithm.



DOI: 10.1002/wcm

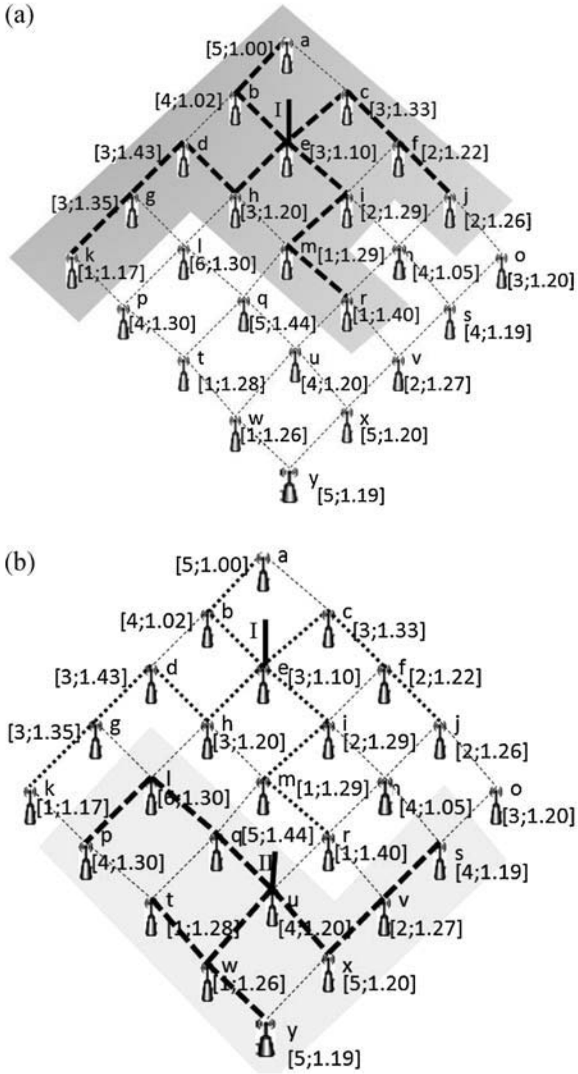


Fig. 4. An example of backhaul assignment. The symbol inside each oval denotes identifier/traffic requirements; the number to the lower right denotes the installation cost (e.g., [5;1.00] denotes that a backhaul requested 5 units of flow and cost 1.00 units). The traffic requirements and installation costs for this example were generated randomly from ranges of 1 to 5 and 1 to 1.6, respectively. The total requested flows, the estimated number of backhauls, and the estimated total flow per backhaul were 78, 3.25, and 23.95, respectively. Here, two backhauls: I and II were assigned. Note that the nodes that have been marked (visited) are filled with lines in the graphic above (the symbols and their meaning are shown in Figure 1). (a) The first backhaul assignments, (b) the second backhaul assignments.

## 3.2. TLBF Routing Algorithm

To solve the TLBF routing problem, we use greedy load-balanced routing (GLBR), shown in Algorithm 2, that adds one link and connects one TAP to one of the given backhauls per iteration (line no. 1–5). In detail, all link costs are initially set to infinity (line no. 2) and all TAPs are marked as FALSE (line no. 3). Subsequently, the incoming link costs of all backhauls are set to adjacent node's traffic $\gamma_v$ (line no. 4). Accordingly, we assign to the forest dominant set $T$ the link with the lowest cost, the minimum number of out-degrees, and the least current total traffic flow to the backhaul links (line no. 11–13). Once this link has been selected, the cost of the out-degree is set to the cost of the previous

**Algorithm 1** WBA ($\varphi$, $V$, $L$)

**Require:** $G = (V, L)$ (a directed graph, where $v \in V$, and $(u, v) \in L$) and the budget $\varphi$.
**Ensure:** The assigned backhaul $b \in B$.

1: CALCULATE the average flow backhaul installation cost, $c_a = \sum_{b \in B} \phi_b / |B|$;
2: $n := \varphi / c_a$;
3: **for** all vertices $v$ **do**
4: $\quad v.marked :=$ FALSE;{set initial value of the tracing node $v$}
5: $\quad T := \{\}$;
6: $\quad$CALCULATE the aggregated flow $\beta_v$ within $H_v$ hop adjacent TAPs;
7: $\quad w_v = \beta_v / \phi_v$;
8: $\quad$CALCULATE the total system flow $t_f$;
9: **end for**
10: $t_b := t_f / n$; {Calculate the expected flow of each candidate backhaul}
11: $t_c := 0$;
12: **while** $t_c < \varphi$ **do**
13: $\quad$SELECT the highest weight, $w_v$, as an assigned backhaul;
14: $\quad v.marked :=$ TRUE;
15: $\quad T := T \cup v$;
16: $\quad t_c := t_c + \phi_v$;
17: $\quad t_a := 0$; {set the initial aggregated flow to 0}
18: $\quad$**while** $t_a < t_b$ and if an adjacent node with $u.marked =$ FALSE **do**
19: $\quad\quad$FIND the minimum link flow of the adjacent TAP $u$ that connects to the minimum aggregate branch flow via TAP $v$ of the current selected backhaul and $u.marked =$ FALSE;
20: $\quad\quad u.marked :=$ TRUE;
21: $\quad\quad t_a := t_a + \gamma_v$; {the amount of traffic flow transmitted via link $(u, v)$}
22: $\quad$**end while**
23: $\quad$UPDATE the aggregated flow, $\beta_v$, within the $H_v$ hop adjacent TAPs which is marked as FALSE;
24: $\quad w_v := \beta_v / \phi_v$;
25: **end while**



DOI: 10.1002/wcm

node plus $\gamma_v$, balancing the traffic flow for each branch (line no. 16–18). The aggregated flow of a node $v$ that belongs to the selected path is also increased by $\gamma_v$ (line no. 19–21). The backhaul aggregated flow $\beta_b$ is also calculated (line no. 22). The above procedures are repeated until all nodes have been marked and assigned to the forest dominant set $T$.

In the code, the 'while-loop' beginning on line no. 10 ensures that each node not included in a backhaul is selected to fulfill Equation (2), to connect to one backhaul. Line no. 11–15 restrict selection to one previous node for each TAP, fulfilling Constraints (5), (6), and (8). Line no. 16–18 and 19–21 increase the link cost and capacity per iteration to fulfill the fairness Constraints (17) and (18). Figure 5 shows an example of nodes in a grid topology with two backhauls. As a result, the objective value is 46, $\alpha' = 0.93$, and $\alpha'' = 0.998$.

**Algorithm 2** GLBR ($B$, $V$, $L$)

**Require:** $G = (V, L)$ (a weighted directed graph where $v \in V$, and link $(u, v) \in L$) and a set of backhauls $B$, where $b \in B$.
**Ensure:** The nodes of the routing tree are included in the dominant set $T$. The variable, $v.pred$, marks the previous node of each relay node, $v$, to a backhaul, $b$.

1: **for** all vertices $v$ **do**
2:    $v.SP := INFINITE$;
3:    $v.mark := FALSE$;
4:    link cost $(s, v) = \gamma_s$; {where $v$ is neighbor of node $s$.}
5: **end for**
6: **for** all backhaul b **do**
7:    $T := T \cup b$;
8:    $b.pred := NULL$;
9: **end for**
10: **while** any one node $v$ is not in $T$ **do**
11:    **for** each node $u$ in $T$ **do**
12:       FIND the minimum cost node $v$ that is not included in the set $T$. Here, we check the capacity Constraint (13) around the path $p$. If more than two links $(u, v)$ have the same cost, we select the minimal branch aggregated flow of node $u$ and the minimal number of out-degree of node $v$.
13:    **end for**
14:    $T := T \cup v$;
15:    $v.pred := u$;
16:    **for** node $u$, which is not in $T$, have the direct link to $v$ **do**
17:       LET link cost $(v, u)$ = link cost $(v, u)$ + $u$;
18:    **end for**
19:    **while** $v.pred$ is not a backhaul **do**
20:       INCREASE the link cost to $\gamma_v$ along the selected path from node $v$.
21:    **end while**
22:    CALCULATE the backhaul aggregated flow $\beta_b = \beta_b + \gamma_v$;
23: **end while**

## 3.3. A Distributed TLBF Routing Algorithms

The above centralized algorithm is described, but it could be converted into a distributed algorithm and used in distributed environments. Since the backhauls as long as build up, it would not be changed immediately. Thus, we focus on the distributed routing algorithm. The algorithm is based on the centralized algorithm with piggyback message exchange and iteration-by-iteration to get the near-optimal distributed solution. The details are described as follows:

**Initial procedures.** Once the backhaul has been assigned, the routing path is changed immediately to adapt to the requirements of the traffic load. To solve the backhaul selection and routing path problem, we enlist our proposed algorithm GLBR, which is based upon the concept of three-way handshaking. The procedures are:

**Step 1.** Initially, the routing procedure proceeds from the node with a backhaul, which passes to its neighboring nodes the message (including information about the link capacity $C_{(u,v)}$ and aggregated traffic load on the backhaul and its branches). The receiving TAP node selects the lightest traffic load and responds to that backhaul.

**Step 2.** The backhaul then sends the ACK message back to the TAP node in its cluster, and sends the NACK message to the nodes not included in this round.

**Step 3.** Once the TAP node has received the ACK message, then for node $s$, the decision variables $g_{sb}$, $x_p$, and a number of the links $y_{(u,v)}$ are determined. Additionally, the backhaul traffic $\beta_b$ and the branch load $f_{(u,b)}$ are calculated.

**Step 4.** Steps 1–3 are repeated for the other unselected nodes. Note that the process node is changed to the leaf nodes of each branch. Based on the data sent in the message, the decision as to which branch is joined is determined at the node. The traffic requirement value $\gamma_s$ is




DOI: 10.1002/wcm

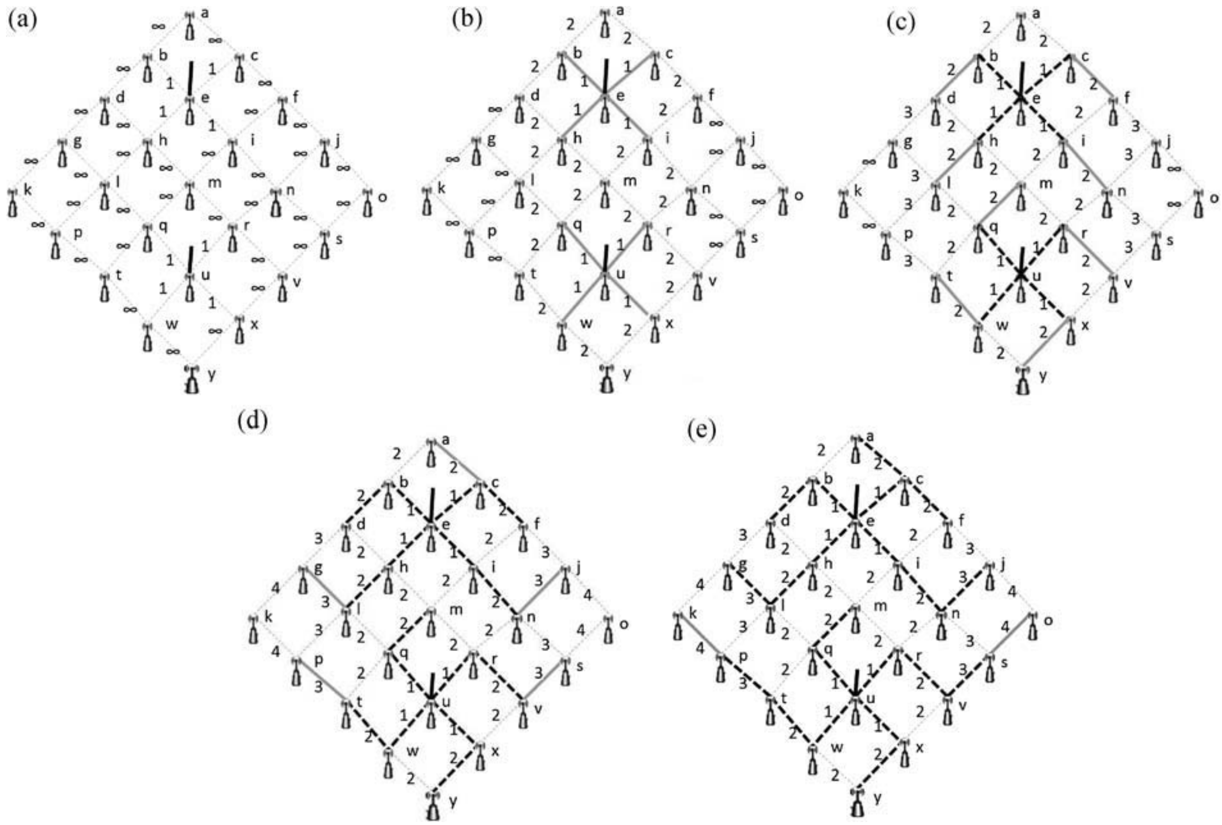


Fig. 5. A simple topology with two backhauls (i.e., backhaul I attached to node $e$ and backhaul II attached to node $u$) to illustrate how the proposed GLBR algorithm works: the traffic requirement $\gamma_s$ is set to 1 unit/s. (The symbols and their meaning are shown in Figure 1.) (a) $e$ and $u$ are included in the set $T = e, u$ and all links are set to $\infty$, but the adjacent link cost of each backhaul is updated to $\gamma_s$. (b) All links adjacent to backhauls are selected and their costs are increased $\gamma_s$. (c) Links with the lowest costs based on the aggregated flow of their backhaul branches are selected; that is, nodes $y$, $v$, $t$, $m$, $n$, $l$, $f$, and $d$ are sequential selected. The relative link costs, branches, and backhauls' aggregated flows are updated. (d) The previous procedure is repeated to determine the minimal cost and minimal branch capacity. Nodes $a$, $g$, $j$, $p$, and $s$ are selected. (e) The remaining nodes $k$ and $o$ are selected

.

transmitted along the path to the corresponding backhaul.

**Step 5.** Each intermediate node fields the lightest requirement to the previous node until the backhaul has been reached. Finally, the backhaul replies to the ACK message with information about the suitable node for its branch and NACK to other nodes. The steps are repeated, iteration upon iteration, until all nodes and paths have been assigned.

**Routing maintenance.** When the traffic requirements of our wireless network change, the routing tables require maintenance to adjust for these developments. When there exists a traffic anomaly, such an irregularity could be detected at the backhaul, which could initiate rebalancing, or at any intermediate node, which would then send a message to the backhaul to start rebalancing procedures. Policies are needed for two conditions: firstly, when the traffic load is over the given threshold for only a short time; and secondly, when the threshold is exceeded for a long period. The latter is a state requiring global balancing, a procedure which makes use of the five steps given previously.

## 4. LR-Based Solutions

First used to solve large-scale integer programming (IP) problems in the 1970s, an LR-based approach [23] exploits the fundamental structure of optimization problems and provides a flexible solution strategy for IP problems. By relaxing certain complex constraints and by using Lagrangean multipliers with an objective function, optimally solvable stand-alone sub-problems are determined [23,26].



DOI: 10.1002/wcm

Achieving near-optimization via an LR-based approach is an NP-complete problem [6,9]. We have developed heuristics to solve this problem in a reasonable amount of time. An LR-based approach determines the LB, used to quantify and evaluate the proposed algorithms.

### 4.1. The LR-Based Approach

Prior to the transformation of the primal problem $Z_{\rm IP}$ given in Section 2 into a Lagrangean dual problem, an iteration of Equation (14) generates Equation (17). By eliminating the denominator from the right-hand side of the equation, Constraint (19) results:

$$\alpha' E_b \sum_{u\in V} f^2_{(u,b)} \le \beta_b^2, \quad \forall b \in B; (u,b) \in L \qquad (19)$$

The aggregated flow over each TAP is less than the adjacent link capacity or the total traffic requirement. This is expressed as

$$0 \le \beta_b \le \min\left(\sum_{u\in V} C_{(u,b)}, \sum_{s\in V}\gamma_s\right) \quad \forall b \in B; (u,b) \in L \qquad (20)$$

Constraint (18) is then modified by allowing $\Gamma = \sum_{b\in B}\beta_b$, denoting the total flow required. By eliminating the denominator from the right-hand side, we arrive at

$$\alpha''|B| \sum_{b\in B}\beta_b^2 \le \Gamma^2 \qquad (21)$$

where $\Gamma$ is also equal to $\sum_{s\in V}\gamma_s$.
Next, the primal problem $Z_{\rm IP}$ is transformed into a Lagrangean dual problem $Z_{\rm LR}$ using Constraints (3), (4), (6), (7), (12), (14), (19), and (21). For a vector of non-negative Lagrangean multipliers, the LR problem is given by:
Objective function

$$\begin{aligned} Z_{\rm LR} = \min\Bigg\{ & \sum_{(u,v)\in L} f_{(u,v)} + \sum_{s\in V}\sum_{b\in B}\mu^1_{sb}[z_{sb}-\eta_b] + \mu^2\left[\sum_{b\in B}\phi_b\eta_b - \varphi\right] \\ & + \sum_{s\in V}\mu^3_s\left(\sum_{b\in B} z_{sb} - \sum_{b\in B}\sum_{p\in P_{sb}} x_p\right) \\ & + \sum_{s\in V}\sum_{b\in B}\sum_{(u,v)\in L}\mu^4_{sbuv}\left(\sum_{p\in P_{sb}} x_p\delta_{p(u,v)} - y_{(u,v)}\right) \\ & + \sum_{(u,v)\in L}\mu^5_{uv}\left(\sum_{s\in V}\sum_{b\in B}\sum_{p\in P_{sb}} x_p\delta_{p(u,v)}\gamma_s - f_{(u,v)}\right) \\ & + \sum_{b\in B}\mu^6_b\left(\beta_b - \sum_{u\in V} f_{(u,b)}\right) \\ & + \sum_{b\in B}\mu^7_b\left(\alpha' E_b \sum_{u\in V} f^2_{(u,b)} - \beta_b^2\right) \\ & + \mu^8\left(\alpha''|B|\sum_{b\in B}\beta_b^2 - \Gamma^2\right)\Bigg\} \end{aligned} \qquad (22)$$

subject to Equations (2), (5), (8), (9), (10), (11), (13), (18), and (20).

The $Z_{\rm LR}$ problem is then decomposed into six independent and solvable sub-problems, described in the Appendix, the summation of which determines the LB. The value of our proposed algorithm and the follow-up primal feasible solution give us an upper bound (UB) for the problem $Z_{\rm IP}$. The distance between the tightest LB and the UB, computed by (UB − LB) / LB × 100%, gives the degree of optimality of the problem solution.

Several methods can be used to solve the Lagrangean dual problems. One of the most popular is the subgradient method [27], which we employed here. Let the decision variable vectors ($x_p$, $z_{sb}$, $\eta_b$, $y_{(u,v)}$, $f_{(u,v)}$, and $\beta_b$) be subgradients of the Lagrangean dual problem. Then, to derive iteration $k+1$ of the subgradient optimization procedures, the multiplier vector, $\pi^k = (\mu^1_{sb}, \mu^2, \mu^3_s, \mu^4_{sbuv}, \mu^5_{uv}, \mu^6_b, \mu^7_b, \mu^8)$ is updated, which gives us $\pi^{k+1} = \pi^k + t^k g^k$. The step size $t^k$ is determined by $t^k = \sigma \cdot \left(Z^h_{\rm IP} - Z_D(\pi^k)\right)/\|g^k\|^2$, where $Z^h_{\rm IP}$ is the primal objective function value for a heuristic solution (an UB on $Z_{\rm IP}$), and $\sigma$ is a constant, $0 < \sigma \le 2$.

Table I shows that the time complexity with a maximum number of iterations $n$ and for a time complexity of $O(n|B||V|^3)$, is dominated by sub-problem (SUB3), which is solved by the minimum sum of the Bellman–Ford algorithm for all nodes.



DOI: 10.1002/wcm

Table I. Time complexity of backhaul assignment and TLBF routing algorithms.

| Sub-problem/procedure | Time complexity | Description |
|---|---|---|
| (SUB1) | $\|V\|\,\|B\|$ | $\|V\|$ sub-problems, each of which finds the minimum weight of the assigned backhauls |
| (SUB2) | $\|V\|\,\|L\|$ | To find the minimum weight of outgoing link $(u, v)$ for $\|V\|$ TAPs |
| (SUB3) | $\|B\|\,\|V\|^3$ | To find the minimum shortest path to the candidate backhauls $\|B\|$ for $\|V\|$ TAPs |
| (SUB4) | $\|L\|$ | To find the minimum value per iteration for $\|L\|$ links |
| (SUB5) | $\|B\|$ | To find the aggregated flow for $\|B\|$ backhauls |
| (SUB6) | $\|B\|$ | To find the backhaul assignment for $\|B\|$ backhauls |
| WBA and GLBR | $\|V\|^2 \cdot E_v$ | To assign the backhauls, find the minimal cost of links adjacent to a tree and the aggregated flow of backhauls for $\|V\|$ nodes |
| LRA* | $n\|B\|\,\|V\|^3$ | Iterated $n$ times, which denotes the number of LR iterations, to find the weight of the minimal routing path to the candidate backhauls $\|B\|$ for $\|V\|$ TAPs |

*The term 'LRA' denotes as LR-based approach.

## 4.2. Primal Feasible Solution

A feasible solution is found if the decision variables of the result are satisfied by the constraints of the primal problem $Z_{\mathrm{IP}}$. The LR-based approach and subgradient method both obtain a theoretical LB and provide some hints about a primal feasible solution.

(a) **Backhaul assignment.** When a backhaul remains undetermined, information provided by the LR-based approach is adopted to assist the assignment of the backhaul to the maximum number of connected TAPs. The procedure is illustrated in Figure 6.

**Step 1.** Let the dominant set $B = \{\}$.

**Step 2.** Select the maximum weight value $\sum_{s\in V} \left( \left( \sum_{(u,v)\in L} (\mu^4_{sbuv} + \mu^5_{uv})\,\delta_{p(u,v)} - \mu^3_s \right) \sum_{p\in P_{sb}} x_p \right)$ based on the hints provided by sub-problem (SUB3), and each TAP route to an assigned backhaul $b$.

**Step 3.** If the remaining budget is sufficient to install this backhaul $b$, let $B = B \cup b$, $\eta_b = 1$, and $\varphi = \varphi - c_b$.

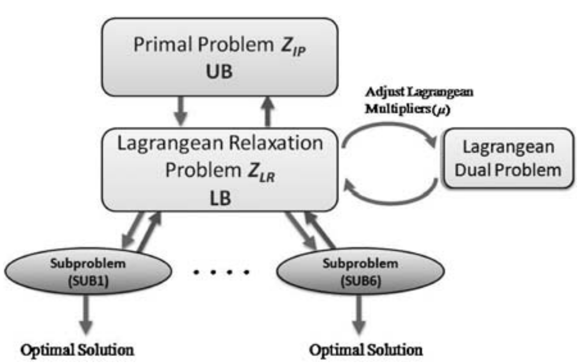


Fig. 6. The procedures of LR approach.

**Step 4.** Repeat Steps 2–3 until all candidate backhauls have been checked.

**Step 5.** After all backhauls have been assigned, the LR-based TLBF heuristic is used to obtain the primal feasible solution for backhaul assignment and TLBF routing problems.

(b) **TLBF routing problem.** Once all the backhauls have been determined, two FI constraints, (17) and (18), are critical to achieving a load-balanced forest. Initially, a node's weight, $\zeta_s = \left(\mu^3_s + \sum_{b\in B} \left(\sum_{(u,v)\in L} \mu^4_{sbuv} + \mu^1_{sb}\right)\right)\gamma_s$ , is adjusted according to the multipliers, after which the current minimal aggregated flow of the backhaul is considered. The minimal aggregated flow over the incoming link to the backhaul is found, as well as the minimal adjacent link's flow to the sub-tree for the specified branch. Accordingly, the flows, aggregated for the backhaul and for the branches, are updated iteration by iteration until all TAPs have been assigned to a backhaul. This procedure is described as follows:

**Step 1.** Adjust the weight of the node $\zeta_s = \left(\mu^3_s + \sum_{b\in B} \left(\sum_{(u,v)\in L} \mu^4_{sbuv} + \mu^1_{sb}\right)\right)\gamma_s$.

**Step 2.** Find the current minimal aggregated flows of the backhauls $b$, the minimal aggregated flow of the incoming link to the backhaul and the minimal weight of the adjacent links for the branch by the GLBR algorithm (shown as Algorithm 2 in Section 3) using the new weights from Step 1.

**Step 3.** Update the aggregated flow of the backhaul and its adjacent branches.

**Step 4.** Repeat Steps 2–3 until all TAPs have been assigned to a backhaul.

Accordingly, the procedures of LR-based approach shown in Figure 6 obtain a near-optimal solution $Z_{\mathrm{IP}}$.




DOI: 10.1002/wcm

Table II. The relative experiment parameters.

| Parameters | Value | Parameters | Value |
|---|---|---|---|
| Maximum number of iterations ($n$) | 10 000 | Maximum number of nodes | 150 |
| Improvement threshold | 100 | Initial scalar of step size ($\sigma$) | 2 |
| LR converge degree | 200 | Fairness index ($\alpha'$ and $\alpha''$) | 0.9 |
| All initial multipliers (i.e., $\mu^1_{sb}, \ldots, \mu^8$) | 0 | Power range | 1 |

The initial parameters are listed as Table II. Six sub-problems are optimally solved to obtain the LB. Once heuristics have obtained the UB, Lagrangean multipliers adjustments are made to the parameters that are relative to LR. The greatest possible time complexity is $|B|\,|V|^3$ per iteration.

### 4.3. A Distributing Adjusting Procedure

A distributed procedure for multiplier adjustment is discussed. Prior to running the procedures of multiplier adjustment, our GLBR enables us to obtain the primal feasible solution (i.e., UB). The subgradient value and the step size $t^k$ are required for each multiplier adjustment. The subgradient values (i.e., $z_{sb} - \eta_b$, $\sum_{b\in B} c_b\eta_b - \varphi$, $z_{sb} - \sum_{p\in P_{sb}} x_p$, $\sum_{p\in P_{sb}} x_p\delta_{p(u,v)} - y_{(u,v)}$, $\sum_{s\in V}\sum_{b\in B}\sum_{p\in P_{sb}} x_p\delta_{p(u,v)}\gamma_s - f_{(u,v)}$, $\sum_{u\in V} f_{(u,b)} - \beta_b$, $\alpha' E_b \sum_{u\in V} f^2_{(u,b)} - \beta^2_b$, and $\alpha''|B|\sum_{b\in B}\beta^2_b - \Gamma^2$) can be calculated at each node or at the node with backhaul (i.e., the current selected backhaul is the CH). The calculation of the step size $t^k$ requires the aggregation of the subgradient values, which are then corrected by the backhaul and exchanged with other backhauls via a wired line network. The Lagrangean value for each sub-problem is then sent to the current backhaul. Accordingly, the step size $t^k$ is calculated by the backhauls, and then the values are broadcasted to its corresponding service TAP nodes. The multiplier is adjusted at each TAP node in preparation for the next iteration.

The detailed centralized and decentralized procedures for the sub-problems of backhaul selection and the routing assignment Lagrangean dual problem are shown in the Appendix. The solutions by which the Lagrangean value and square subgradient values are obtained for each sub-problem are described as follows:

**Step 1.** Initialize. Assign a nonnegative value to each multiplier. Set the iteration counter, $k$ , to 0.

**Step 2.** Examine stopping criteria. If the number of iterations has reached the given value, then stop; otherwise, proceed with Step 3.

**Step 3.** Solve via LR. The distributed algorithms described above are used to solve sub-problems (SUB1)–(SUB6). For each source node $s$, the solution determines both the backhaul that is to be selected and the routing path. For each backhaul, it determines the branch aggregated flow and backhaul aggregated flow. For each link $(u, v)$, an estimate is made at each node $v$ of its incoming aggregated flow and determines which link will be selected for transmission.

**Step 4.** Adjust multipliers:

a. For each source node $s$, calculate $\mu^{1k}_{sb}$ and $\mu^{3k}_s$. For each backhaul $b$, calculate $\mu^{2k}$, $\mu^{6k}_b$, $\mu^{7k}_b$, and $\mu^{8k}$. For each tail of link $(u, v)$, each node $v$ calculates $\mu^{4k}_{sbuv}$ and $\mu^{5k}_{uv}$. The detailed procedures are described above.

b. $k \leftarrow k + 1$.

c. Go to Step 2.

## 5. Evaluation and Experiment Results

We generated a set of nodes, $V$, for each of two cases. In the first case, we used a grid-based topology. In the second, we randomly distributed TAP nodes within a specific area. The maximum transmission range of each node in either case was set to 1. The relative parameters we used are given in Table II. When nodes are randomly deployed, the connectivity of a network is simply a function of the average number of neighbors. The following two conditions are compared: (i) variations in the numbers of nodes; and (ii) variations in the budgets.

The LR-based approach ensures that results lie in the gap that falls between the UB and the LB, obtained from the Lagrangean dual problem. For that reason, to enhance the quality of our solutions, the gap was kept as small as possible. Figure 7 shows the result of an LR experiment, where the LB reached a near-optimal value after approximately 1200 iterations. The gap in this experiment was about 3.5%. As we were able to relax some of the constraints in this experiment, we were



DOI: 10.1002/wcm

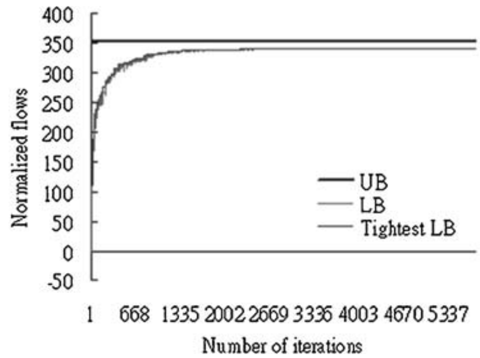


Fig. 7. An example of LR experimental results, in which optimization is bounded between the UB and the tightest LB, with four given backhauls assigned on $11 \times 11$ grid graphs. Here, the gap was equal to 3.5% and $FI = 0.92$.

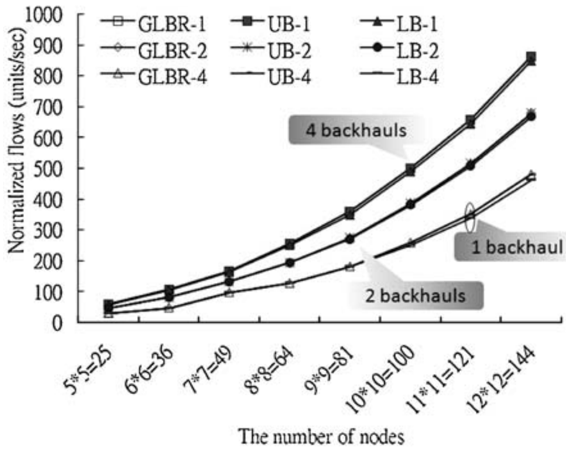


Fig. 8. Case 1) grid-based experimental results where UB-$x$ and LB-$x$ denote the UB and LB with $x$ backhauls, respectively. The gap was less than 5%.

able to get a narrow gap, which meant that our proposed algorithms determined a near-optimal solution.

(a) **TLBF routing algorithm.** The following experiments tested the routing algorithm in networks where the number of backhauls was given. The UB and LB are denoted UB-$x$ and LB-$x$, respectively, where $x$ denotes the number of assigned backhauls. Experimental results for the first case, given 1, 2, or 4 backhauls, are depicted in Figure 8. The gap was found to be less than 5%. In a simple case such as this, when the number of nodes increases, the number of hops from a node to reach a given backhaul increases. Normalized network flow increases as the number of nodes increases. Thus, the minimum normalized flows increases, and the curve tends to

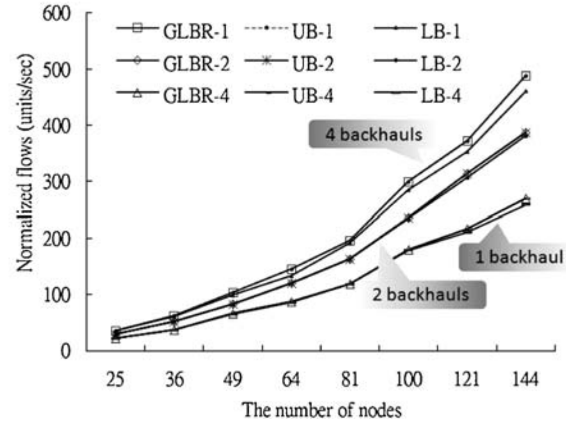


Fig. 9. Case 2): Random-deployment results for the UB and the LB. In this experiment, the gap is less than 10%.

increase exponentially. However, when the number of backhauls is increased, the normalized flows are reduced by the proportion of flows per backhaul decreases.

Figure 9 shows the results from the second case, the random-based experiment, with 1, 2, or 4 backhauls. The gap for this experiment case was about 10%, which is larger than the grid-based gap due to different numbers of degrees for each backhaul and to relaxation of FI constraints. This case showed that our proposed algorithm was able to obtain the objective value within 10% of the optimal value. In this experiment, as the number of nodes increased, the flows also increased. However, shorter paths to backhauls were found by some TAPs. As a result, the aggregated flow was less than the flow in a grid network. Traffic in the network is reduced when TAPs have more choices when routing to the egress.

(b) **Backhaul assignment.** The proposed WBA and GLBR algorithms along with LR-based heuristics compare favorably with the Lowest IDentifier (LID) [28] and HD algorithms [8]. The LID algorithm uses criteria to allow a TAP to become elected as a backhaul if it has the minimum ID of nodes within a neighborhood of $H_s$ hops. The HD algorithm elects the TAP that has the highest degree in its $H_s$-hop's neighboring nodes as a backhaul. Once the backhauls are elected, the GLBR algorithm is adopted to solve the TLBF problem for these backhaul assignment algorithms. Figures 10–12 show Experiment results for budgets of 3, 5, and 7 price units that demonstrate the quality of our proposed algorithm WBA.



DOI: 10.1002/wcm

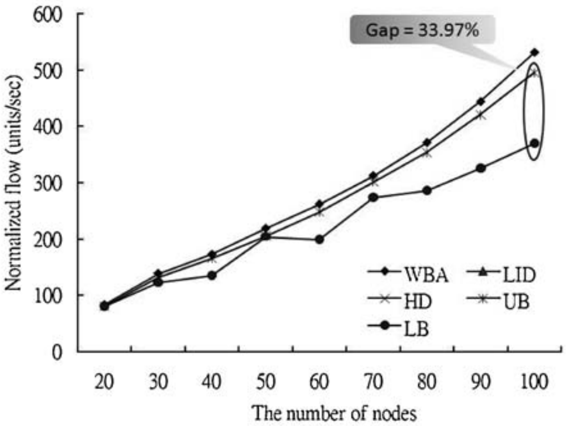


Fig. 10. Results for $\varphi = 3$. There were no feasible solutions for the LID and HD algorithms. Thus, there are no experimental results for them in this case.

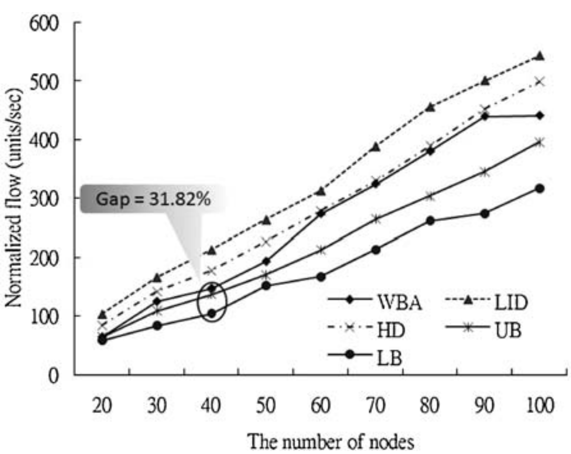


Fig. 11. Results for $\varphi = 5$.

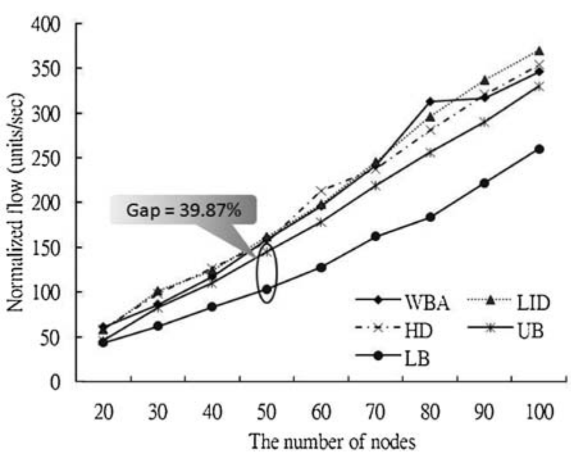


Fig. 12. Results for $\varphi = 7$.

Figure 10 shows experimental results for $\varphi = 3$. The gap was less than 35%. The LID and HD algorithms do not select backhauls according to gravitation, so the FI was violated in this condition. Though the LID and HD algorithms do not give feasible solutions, our proposed algorithm obtained a feasible solution and lower aggregated flow as it selects backhauls based on the traffic requirement gravitation.

Figure 11 shows experiment results for $\varphi = 5$. The gap was less than 32%. The flow increased linearly. Lower aggregated flows were seen for WBA than for the GLBR algorithm with given backhauls, demonstrating that it reduces the network traffic flow. The LR-based approach is better than the proposed GLBR plus WBA algorithms by 20%, however, it should be mentioned that the time complexity is higher than that of the proposed algorithms. Use of the proposed algorithms resulted in a 10% improvement in the performance.

Figure 12 shows the experimental results for $\varphi = 7$. The gap was less than 40%. Due to increases in the budget, the gap was larger than that for previous experiments. The performance of the HD algorithm performed approximately as well as our proposed algorithms alone, but LRA obtained more than a 10% improvement.

The UB is obtained by the feasible primal feasible solution. The solution could be any algorithm that fulfills the constraints in primal problem. Its value is higher than the real optimal objective value. Oppositely, the LB is calculated by the Lagrangean dual-mode problem, where some constraints are relaxed. In addition, the dual-mode problem is divided into many sub-problems. Thus, its value is lower than optimal value. Only when the solution is close to the optimal solution, the LB is close to optimal value. In other words, it means the subgradient value is approach to 0. The LB is also affected by the multipliers, which is adjusted by the sub-gradient method toward the optimal solutions.

Based on the above explanation, the LR approach, which provides a serial procedures to solve the optimization problem, could be applied to any schemes to solve the problem or any partial of issues. In this paper, we focus on two major issues: (i) backhaul assignment issue and (ii) load-balanced routing issue.



DOI: 10.1002/wcm

- When (ii) is considered, the backhauls are assigned and then compare how well a routing algorithm to achieve higher performance. Thus, we device our load-balanced scheme to get better performance.
- When (i) is considered, the routing algorithms are fixed with our proposed GLBR scheme and then compare the backhaul assignment schemes so that the comparable bench mark is fair.

However, the gap (the distance of UB and LB) is still different between (i) and (ii) because they have different sets of constraints, decision variables, and given parameters. The solution schemes are also different to cause the differential UB and LB.

## 6. Conclusions

The two algorithms proposed in this paper, WBA and GLBR, in conjunction with an LR-based approach, solve the 'backhaul assignment' and 'TLBF' problems. The WBA algorithm handles backhaul assignment, while GLBR algorithm handles fairness and capacity limits. Using these algorithms yields not only the minimum objective function value and balances branch loads of the backhauls, but also achieves nearly equivalent amounts of flow between backhauls. The proposed algorithms were evaluated by comparisons with the LB, obtained from an LRA. The TLBF experiment results demonstrated that the proposed algorithms arrive at near optimal solutions with gaps of less than 5% for a grid-based topology, and less than 10% for a random-based topology. The backhaul assignment plus TLBF experiment results demonstrate that the WBA plus GLBR algorithms and LRA obtain 10 and 30% improvement with gaps of less than 40%. The greatest time complexity for these approaches is $O(n|B|\,|V|^3)$.

In this paper, we focus on the experiments to the major issue with LR approach. This approach focuses on the heuristic algorithms. The quantities of the solutions are evaluated by the duality gaps. The reason random graphs were used because this is one of most general way to compare how better the proposed algorithms. As the algorithms performed well in these graphs, actual wireless network graphs could be simulated and evaluated. Future work can integrate the proposed method into simulation tools, such as NS2, to determine such advanced solutions.

## Appendix : The Six Sub-Problems of Backhaul Assignment Plus TLBF Routing

Each of the following six sub-problems, generated from the Lagrangean dual problem $Z_{\mathrm{LR}}$, is related to a decision variable. The distributed solutions by which the Lagrangean value and square subgradient values are obtained for each sub-problem are described as follows.

**Sub-problem (SUB1)**, related to decision variable $z_{sb}$.
Objective function:

$$Z_{\mathrm{SUB1}} = \min\left\{\sum_{s\in V}\sum_{b\in B}\left(\mu_{sb}^{1}+\mu_{s}^{3}\right)z_{sb}\right\} \quad \text{(SUB1)}$$

subject to Equation (2).

This sub-problem, related to $z_{sb}$, is further decomposed into $|V|$ sub-problems, solved by first sorting the weights $(\mu_{sb}^{1}+\mu_{s}^{3})$ for each node $s$, and then by selecting the minimum weight, thus setting $z_{sb}$ equal to 1 according to Equation (2). The minimum objective value of sub-problem (SUB1), summarized for all $s\in V$.

For iteration $k$, let $\mu_{sb}^{1k}$ be the multiplier that represents the relation between node $s$ and backhaul $b$, and let $z_{sb}^{k}$ be the solution to sub-problem (SUB1). To select a backhaul $b$ for node $s$, the multiplier $\mu_{sb}^{1k}$ must be updated to $\mu_{sb}^{1(k+1)}$, as determined by $\mu_{sb}^{1(k+1)} = \mu_{sb}^{1k} + t^{k}\left(z_{sb}^{k} - \sum_{p\in P_{sb}} x_{p}^{k}\right)$. Subsequently, the node selects the minimum value that allows $z_{sb}^{k}$ to be set to 1, and it then transmits this value to its current backhaul. In iteration $k$, the decision variables $z_{sb}^{k}$ and $x_{p}^{k}$ are recorded at node $s$. The step size $t^{k}$ was received from the backhaul.

Let $coef_{sb}^{k}$ be the subgradient value $z_{sb} - \sum_{p\in P_{sb}} x_{p}$ for node $s$ to backhaul $b$. The decision variable is solved by (SUB3), to be presented in the following paragraphs. Accordingly, this sub-problem is solved at node $s$, which then sends the Lagrangean value $Z_{\mathrm{SUB1}}$ and square subgradient values $coef_{sb}^{1k}$ to the backhaul.

**Sub-problem (SUB2)**, related to decision variable $y_{(u,v)}$.
Objective function

$$Z_{\mathrm{SUB2}} = \min\left\{\sum_{(u,v)\in L}\left(\sum_{s\in V}\sum_{b\in B}\left(-\mu_{sbuv}^{4}\right)\right)y_{(u,v)}\right\} \quad \text{(SUB2)}$$

subject to Equations (8)–(11).

This sub-problem, related to $y_{(u,v)}$, can be further decomposed into $|L|$ sub-problems. The following



DOI: 10.1002/wcm

procedures determine the solution.

**Step 1.** Let $\theta_{(u,v)}$ denote the weight of link $(u, v)$ results in $\theta_{(u,v)} = \sum_{s\in V}\sum_{b\in B} -\mu^4_{sbuv}$.
**Step 2.** Search the minimum weight $\theta_{(u,v)}$ of each incoming link of each node, $v$. We then assign the value of $y_{(u,v)} = 1$ and assign the value of $y_{(u,v)} = 0$ to all other links thus fulfilling Constraints (8)–(11).
**Step 3.** Sort the weight $\theta_{(u,v)}$ for all $y_{(u,v)} = 1$.
**Step 4.** Reset the last minimum weight $y_{(u,v)} = 1$.
**Step 5.** Iterate to calculate the objective value of sub-problem (SUB2).

For iteration $k$, let $\mu^{4k}_{sbuv}$ be the multiplier used for the relation between node $s$ and backhaul $b$ on link $(u, v)$, let $x^k_{sbuv}$ be an array for the recording of the links $(u, v)$ that are used for the routing path that extends from node $s$ to backhaul $b$, and let $y^k_{(u,v)}$ be the solution to sub-problem (SUB2). When an outgoing link for node $u$ is to be selected, the multiplier is then updated from $\mu^{4k}_{sbuv}$ to $\mu^{4(k+1)}_{sbuv}$, as determined by $\mu^{4(k+1)}_{sbuv} = \mu^{4k}_{sbuv} + t^k\big(\sum_{p\in P_{sb}} x_p\delta_{p(u,v)} - y_{(u,v)}\big)$. The variables $\mu^{4k}_{sbuv}$, $x^k_{sbuv}$, and $y^k_{(u,v)}$ were previously recorded at node $u$, and the step size $t^k$ was obtained from the backhaul. After determining the multiplier, for node $u$ the outgoing link $y^k_{(u,v)}$ is set equal to 1 based on the minimum value of $\sum_{s\in V}\sum_{b\in B}\mu^{4k}_{sbuv}$ and in accordance with the Constraints (8)–(10). We let $coef^k_{sbuv}$ be the subgradient value for each link $(u, v)$ on the path from node $s$ to backhaul $b$. Solving the decision variable $x^k_{sbuv}$ is done by means of (SUB3), which is detailed in the following paragraph. Accordingly, this sub-problem (SUB2) is solved at node $u$, which then sends the Lagrangean value $Z_{\mathrm{SUB2}}$ and square subgradient values $coef^{2k}_{sb}$ to the backhaul.

**Sub-problem (SUB3)**, related to decision variable $x_p$.
Objective function

$$Z_{\mathrm{SUB3}} = \min\left\{\sum_{s\in V}\sum_{b\in B}\Big(\sum_{(u,v)\in L}(\mu^4_{sbuv} + \mu^5_{uv}\gamma_s)\delta_{p(u,v)} - \mu^3_s\sum_{p\in P_{sb}} x_p\Big)\right\} \tag{SUB3}$$

subject to Equation (5).

This sub-problem is related the ‘to-be-determined’ path, $x_p$, and can be further decomposed into $|V|$ sub-problems. Each sub-problem is a shortest path problem solved by considering the link weight $\sum_{(u,v)\in L}(\mu^4_{sbuv} + \mu^5_{uv}\gamma_s)\delta_{p(u,v)}$ that subtracts the multiplier value $\mu^3_s$. It can be easily solved by the Bellman–Ford algorithm for each node to one of backhauls $B$. The aggregation of these minimum values of all TAPs, $s$, forms the objective value of sub-problem (SUB3).

The traffic requirement $\gamma_s$ for node $s$ is initially broadcasted to all nodes that are situated within $H_s$ hops, as constrained by Equation (11). For iteration $k$, $\mu^{5k}_{uv}$ is the multiplier for the traffic load on link $(u, v)$, and $x^k_p$ is the solution to sub-problem (SUB3) in iteration $k$ for that path $p$ that is selected to route from node $s$ to the backhaul $b$. In determining the route to the backhaul $b$, the multipliers $\mu^{3k}_{sb}$, $\mu^{4k}_{sbuv}$, and $\mu^{5k}_{uv}$ are adjusted. The values of the multipliers $\mu^{3(k+1)}_{sb}$ and $\mu^{4(k+1)}_{sbuv}$ are found using the procedures that were described in the previous subsection; $\mu^{5(k+1)}_{uv}$ is determined by $\mu^{5(k+1)}_{uv} = \mu^{5k}_{uv} + t^k\big(\sum_{s\in V}\sum_{b\in B}\sum_{p\in P_{sb}} x_p\delta_{p(u,v)}\gamma_s - f_{(u,v)}\big)$. The variables $\mu^{4k}_{sbuv}$, $\mu^{5k}_{uv}$, and $\gamma_s$ are recorded at node $u$ and $\mu^{3k}_{sb}$ is recorded at node $s$. Based on these data, the link cost is calculated according to $\mu^{4k}_{sbuv} + \mu^{5k}_{uv}\gamma_s$ along the links from node $s$ routing to the backhaul $b$. We use the distributed Bellman–Ford algorithm to find the routing path with initial cost $-\mu^{3k}_{sb}$ and maximum $H_s$ hops. The minimum cost of the path is equal to the value of the Lagrangean of sub-problem (SUB3) for node $s$.

Let $\Phi^k_{(u,v)}$ denote the aggregated flow for all selected routing paths and the $coef^{3k}_{uv}$ is the subgradient value $\Phi^k_{(u,v)} - f^k_{(u,v)}$ for a link $(u, v)$ that is on the path from node $s$ to backhaul $b$. The decision variable $f^k_{(u,v)}$ is determined by means of (SUB4) later. Accordingly, the sub-problem is solved at node $s$, which then sends the Lagrangean value $Z_{\mathrm{SUB3}}$ and square subgradient values $coef^{5k}_{uv}$ to the backhaul.

**Sub-problem (SUB4)**, related to decision variable $f_{(u,v)}$.
Objective function

$$Z_{\mathrm{SUB4}} = \min\sum_{(u,b)\in L}\Big(\mu^7_b\alpha' E_b f^2_{(u,b)} - \mu^6_b + \mu^5_{uv} - 1\Big)f_{(u,b)} + \sum_{(u,v)\in L, v\neq b}(1-\mu^5_{uv})f_{(u,v)}\Big) \tag{SUB4}$$

subject to Equation (13).

This sub-problem, related to the aggregated flow $f_{(u,v)}$, may be decomposed into two sub-equations

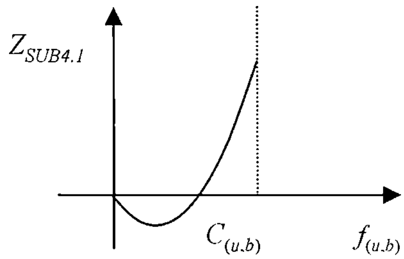


Fig. 13. The objective function convex curve of sub-problem (SUB4.1).

(SUB4.1) and (SUB4.2).

$$Z_{\text{SUB4.1}} = \min \sum_{(u,b)\in L} \left(\mu_b^7 \alpha' E_b f_{(u,v)}^2 - \left(\mu_b^6 + \mu_{ub}^5 - 1\right) f_{(u,v)}\right) \quad \forall (u,v) \in L, b \in B \qquad \text{(SUB4.1)}$$

$$Z_{\text{SUB4.2}} = \min \left\{ \sum_{(u,v)\in L} \left((1-\mu_{uv}^5) f_{(u,v)}\right) \right\} \quad \forall (u,v) \in L, v \notin B \qquad \text{(SUB4.2)}$$

The curvature of Sub-problem (SUB4.1), shown in Figure 13, is a quadratic equation. We set the decision variable $f_{(u,v)} = 0$ when the coefficient $\left(\mu_b^6 + \mu_{ub}^5 - 1\right) \leq 0$; otherwise, we set the decision variable $f_{(u,v)} = \left(\mu_b^6 + \mu_{ub}^5 - 1\right)/2\mu_b^7 \alpha' E_b$ for any backhaul $b$.

Sub-problem (SUB4.2) is a simple equation. We are able to directly set the $f_{(u,v)} = 0$ when the coefficient $\left(1 - \mu_{uv}^5\right) > 0$; otherwise, $f_{(u,v)} = C_{(u,v)}$. The aggregation of these minimum values of all links forms the objective value of sub-problem (SUB4).

For iteration $k$, $\mu_b^{6k}$ and $\mu_b^{7k}$ are the multipliers for aggregated traffic on backhaul branch $(u, b)$ and on backhaul $b$, respectively. Let $f_{(u,v)}^k$ be the solution to sub-problem (SUB4) in iteration $k$ that the amount of traffic on link $(u, v)$. To determine the aggregated flow $f_{(u,v)}^k$, the multipliers $\mu_{ub}^{5k}$, $\mu_b^{6k}$, and $\mu_b^{7k}$ are adjusted. The procedures for updating the multiplier $\mu_{ub}^{5(k+1)}$ were described in the previous subsection. The value of $\mu_b^{6(k+1)}$ is determined by $\mu_b^{6(k+1)} = \mu_b^{6k} + t^k\left(\sum_{u\in V} f_{(u,b)} - \beta_b\right)$, and $\mu_b^{7(k+1)}$ is determined by $\mu_b^{7(k+1)} = \mu_b^{7k} + t^k\left(\alpha' E_b \sum_{u\in V} f_{(u,b)}^2 - \beta_b^2\right)$. Both of these multipliers $\mu_b^{6k}$ and $\mu_b^{7k}$ are recorded at backhaul $b$. For each link adjacent to the backhaul, the link aggregate flow is calculated by (SUB4.1); for all other links, it is calculated by (SUB4.1). A node $u$ that is adjacent to the backhaul passes the value $\mu_{ub}^{5k}$ to the backhaul. Accordingly, the backhaul $b$ sets the decision variable $f_{(u,b)} = 0$ when the coefficient $(\mu_b^{6k} + \mu_{ub}^{5k} - 1) \leq 0$; otherwise, the decision variable is set $f_{(u,b)} = (\mu_b^{6k} + \mu_{ub}^{5k} - 1)/2\mu_b^{7k}\alpha' E_b$. Any node $u$ that is not adjacent to the backhaul sets the decision variable $f_{(u,v)} = 0$ when the coefficient $(1 - \mu_{uv}^5) > 0$; otherwise, $f_{(u,v)}$ is set equal to $C_{(u,v)}$. The aggregation of these minimum values of all links $(u, v) \in L$ forms the objective value of sub-problem (SUB4).

Let $coef_b^{6k}$ be the subgradient value $\sum_{u\in V} f_{(u,b)}^k - \beta_b^k$ and $coef_b^{7k}$ be the subgradient value $\alpha' E_b \sum_{u\in V} f_{(u,b)}^k - \beta_b^k$ for a link $(u, b)$ adjacent to the backhaul $b$. The decision variable $\beta_b^k$ is solved by (SUB5) in the following paragraph. Accordingly, this sub-problem is solved at node $u$ and at the backhaul $b$. Node $u$ subsequently sends the Lagrangean value $Z_{SUB4}$ and the square subgradient values $coef_{uv}^{5k}$ to the backhaul.

**Sub-problem (SUB5)**, related to decision variable $\beta_b$. Objective function

$$Z_{\text{SUB5}} = \min \left\{ \sum_{b\in B} \left(\mu_b^6 \beta_b + (\mu^8 \alpha'' |B| - \mu_b^7)\beta_b^2\right) \right\} \qquad \text{(SUB5)}$$

subject to Equation (20).

This sub-problem is related to the aggregated flow of backhaul $b$, $\beta_b$, and can be further decomposed into $|B|$ sub-problems. The aggregated flow of a backhaul $b$ is less than the link capacity times the number of branches, which means the maximum value of $\beta_b$ is less than $\sum_{(u,b)\in L} C_{(u,b)}$. The curve of sub-problem (SUB5), shown in Figure 14, is a quadratic equation that enables us to set the decision variable $\beta_b = 0$ when the coefficient $\mu^8\alpha''|B| - \mu_b^7 > 0$; otherwise, it is set to the maximal value $\sum_{(u,b)\in L} C_{(u,b)}$ to get the minimal objective value.

For iteration $k$, let $\mu^{8k}$ be the multipliers for aggregated traffic of all backhaul $b$. Let $\beta_b^k$ be the solution to sub-problem (SUB5) that is the aggregated traffic flow of backhaul $b$. In order to calculate the aggregated flow $\beta_b^k$, the multipliers $\mu_b^{7k}$ and $\mu^{8k}$ are adjusted. The updated multiplier $\mu^{8(k+1)}$ is determined by $\mu^{8(k+1)} = \mu^{8k} + t^k(\alpha''|B| \sum_{b\in B} \beta_b^2 - \Gamma^2)$. The multiplier $\mu^{8k}$ is recorded at backhaul $b$, to be used to correct the square of the traffic requirements of all source nodes and the square of the aggregated traffic from all backhauls. Next, the minimal sub-objective value



DOI: 10.1002/wcm

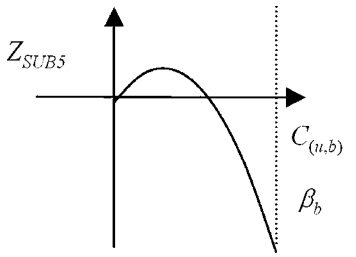


Fig. 14. The objective function's convex curve of (SUB5).

is found at each backhaul based on the solution for (SUB5). The aggregation of all of the minimum values for backhaul $b$ forms the objective value of sub-problem (SUB5). Let $coef_b^{8k}$ be the subgradient value $\alpha''|B|\sum_{b\in B}\beta_b^2 - \Gamma^2$. The Lagrangean value $Z_{\text{SUB5}}$ and the square subgradient values $coef_b^{8k}$ are sent to the backhaul.

**Sub-problem (SUB6)**, related to decision variable $\eta_b$. Objective function

$$Z_{\text{SUB6}} = \min\left\{\sum_{b\in B}\left(\mu^2\phi_b - \sum_{s\in V}\mu_{sb}^1\right)\eta_b\right\} \tag{SUB6}$$

subject to $\eta_b$ is a 0-1 decision variable.

Related to the backhaul assignment $\eta_b$, this sub-problem is further decomposed into $|B|$ sub-problems. Each sub-problem is solved by sorting the weight $\mu^2 c_b - \sum_{s\in V}\mu_{sb}^1$, then selecting the minimum weight to set the $\eta_b$ equal to 1 before the total cost is larger than the budget. For other unselected backhaul $b$, $\eta_b$ is set to 0.

For iteration $k$, let $\mu^{2k}$ be the multipliers for budget of the network. Let $\eta_b^k$ be the solution to sub-problem (SUB6) that is whether a backhaul $b$ is assigned. In order to calculate the value $\eta_b^k$, the multipliers $\mu_{sb}^{1k}$ and $\mu^{2k}$ are adjusted. The updated multiplier $\mu_{sb}^{1(k+1)}$ is determined by $\mu_{sb}^{1(k+1)} = \mu_{sb}^{1k} + t^k(z_{sb} - \eta_b)$. The updated multiplier $\mu^{2k}$ is determined by $\mu^{2(k+1)} = \mu^{1k} + t^k(\sum_{b\in B}\phi_b\eta_b - \varphi)$. The multipliers $\mu_{sb}^{1k}$ and $\mu^{2k}$ is recorded at backhaul $b$, to be used to correct the square of the backhaul selections of all source nodes and the square of the build-up costs from all backhauls. The weight $\mu^2 c_b - \sum_{s\in V}\mu_{sb}^1$ is calculated in each node and exchange with neighbor node to compare the maximum weight and selected as backhaul within $H_b$ hops and the value of $\eta_b^k$ is set to be 1, otherwise 0.

For subgradient method, let $coef_{sb}^{1k}$ be the subgradient value $z_{sb} - \eta_b$ and $coef^{2k}$ be the subgradient value $\sum_{b\in B}\phi_b\eta_b - \varphi$. Both $coef_{sb}^{1k}$ and $coef^{2k}$ are calculated in each candidate backhaul.

## References


1. Xiao Y, Leung KK, Pan Y, Du X. Architecture, mobility management, and quality of service for integrated 3G and WLAN networks. *Wireless Communications and Mobile Computing* 2005; **5**(7): 805–823.
2. Urban J, Wisely D, Bolinth E, Neureiter G, Liljeberg M, Valladares TR. BRAIN—an architecture for a broadband radio access network of the next generation. *Wireless Communications and Mobile Computing* 2001; **1**(1): 55–75.
3. Akyildiz IF, Wang X, Wang W. Wireless mesh networks: a survey. *Elsevier Computer Networks* 2005; **47**(4): 445–487.
4. Karrer R, Sabharwal A, Knightly EW. Enabling large-scale wireless broadband: the case for TAPs. *ACM Computer Communication Review* 2004; **34**(1): 27–32.
5. Bruno R, Conti M, Gregori E. Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine* 2005; **43**(3): 123–131.
6. Amis AD, Prakash R, Vuong THP, Huynh DT. Max-min d-cluster formation in wireless ad hoc networks. In *Proceedings of IEEE INFOCOM* 2000; 32–41.
7. Baker DJ, Ephremides A. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications* 1981; **COM-29**(11): 1694–1701.
8. Parekh AK. Selecting routers in ad-hoc wireless networks. In *Proceeding of the SBT/IEEE ITS* 1994; 420–424.
9. Mehrotra A, Trick MA. Cliques and clustering: a combinatorial approach. *Operations Research Letters* 1997; **22**(1): 1–12.
10. Aslam J, Li Q, Rus D. Three power-aware routing algorithms for sensor networks. *Wireless Communications and Mobile Computing* 2003; **3**(2): 187–208.
11. Belding-Royer EM. Hierarchical routing in ad hoc mobile networks. *Wireless Communications and Mobile Computing* 2002; **2**(5): 515–532.
12. Ye J, Papavassiliou S, Tekinay S. An efficient and fair cooperative approach for resource management in wireless networks. *Wireless Communications and Mobile Computing* 2005; **5**(7): 847–861.
13. Chiu C-Y, Chen G-H, Wu E H-K. A stability aware cluster routing protocol for mobile ad hoc networks. *Wireless Communications and Mobile Computing* 2003; **3**(4): 503–515.
14. Johnson D, Maltz D. *Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing*, Chapter 5. Kluwer Academic Publishers: Dordrecht, The Netherlands, 1996.
15. Park VD, Corson MS. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of the IEEE INFOCOM* 1997; **3**: 1405–1413.
16. Perkins CE, Bhagwat P. Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers. In *Proceedings of the ACM SIGCOMM* 1994; 234–244.
17. Perkins CE, Royer EM. Ad hoc on-demand distance vector routing. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications* 1999; 90–100.
18. Chang YL, Hsu CC. Routing in wireless/mobile ad hoc networks via dynamic group construction. *ACM Mobile Networks and Applications* 2000; **5**(1): 27–37.
19. Hou T-C, Tsai T-J. An access-based clustering protocol for multihop wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications* 2001; **19**(7): 1201–1210.
20. Pearlman MR, Haas ZJ. Determining the optimal configuration of the zone routing protocol. *IEEE Journal on Selected Areas in Communications* 1999; **17**(8): 1395–1414.
21. Dai H, Han R. A node-centric load balanced algorithm for wireless sensor networks. In *Proceedings of IEEE GLOBECOM* 2003; **3**: 548–552.
22. Hsiao PH, Hwang A, Kung HT, Vlah D. Load-balanced routing for wireless access networks. In *Proceedings of IEEE INFOCOM* 2001; 986–995.




DOI: 10.1002/wcm

23. Fisher ML. The Lagrangean relaxation method for solving integer programming problems. *Management Science* 1981; **27**(1): 1–18.
24. Jain R, Hawe W, Chiu D. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *DEC Research Report* TR-301 1984. http://www.cis.ohio-stateedu/jain/papers/ fairness.htm
25. Hardy GH, Littlewood JE, Polya G. *Tchebychef's Inequality*. §2.17 and §5.8 in Inequalities (2nd edn). Cambridge University Press: Cambridge, England, 1988; 43–45 and 123.
26. Ahuja RK, Magnanti TL, Orlin JB. *Network Flows: Theory, Algorithms, and Applications*, Chapters 4 and 16, Prentice-Hall: NJ, 1993.
27. Held M, Wolfe P, Crowder HP. Validation of subgradient optimization. *Math Programming* 1974; **6**(1): 62–88.
28. Ephremides A, Wieselthier JE, Baker DJ. A design concept for reliable mobile radio networks with frequency hopping signaling. In *Proceedings of the IEEE* 1987; **75**(1): 56–73.

## Authors' Biographies

**Frank Yeong-Sung Lin** received his B.S. degree in electrical engineering from the Electrical Engineering Department, National Taiwan University in 1983, and his Ph.D. in electrical engineering from the Electrical Engineering Department, University of Southern California in 1991. After graduating from the USC, he joined Telcordia Technologies (formerly Bell Communications Research, abbreviated as Bellcore) in New Jersey, U.S.A., where he was responsible for developing network planning and capacity management algorithms. In 1994, Professor Lin joined the faculty of the Electronic Engineering Department, National Taiwan University of Science and Technology. Since 1996, he has been with the faculty of the Information Management Department, National Taiwan University. His research interests include network optimization, network planning, network survivability, performance evaluation, high-speed networks, distributed algorithms, content-based information retrieval/filtering, biometrics, and network/information security.

**Yean-Fu Wen** received his M.S. degree from the Department of Information Management, National Taiwan University of Science Technology, Taiwan, R.O.C., in 1998. He finished his Ph.D. study and got a doctoral degree from the Department of Information Management, National Taiwan University in July 2007. In February 2008, he joined the Department of Information Management, Chinese Culture University as an Assistant Professor. His research interests include network planning, performance optimization, and cross-layer technology in next-generation wireless networks. He is a member of IEEE.





DOI: 10.1002/wcm