PAPER *Special Section on Advanced Transfer Technologies for the Next Generation Network*

Fair Bandwidth Allocation and End-to-End Delay Routing Algorithms for Wireless Mesh Networks

Yean-Fu WEN[†,††a)], ***Student Member and*** **Yeong-Sung (Frank) LIN**[†], ***Nonmember***

SUMMARY Quality-of-service (QoS) is essential for multimedia applications, such as video-conferencing and voice over IP (VoIP) services, in wireless mesh networks (WMNs). A consequence of many clients accessing the Internet via the same backhaul is that throughput depends on the number of hops from the backhaul. This spatial bias problem is formulated as a mixed-integer nonlinear programming problem that considers end-to-end delay in terms of gateway selection, least-hop and load-balanced routing, and link capacity constraints. In this paper, we propose a routing algorithm for the network layer and a bandwidth allocation scheme for the medium access control (MAC) layer. The latter achieves fairness in both throughput and end-to-end delay in orthogonal mesh backbone networks with a distributed scheme, thereby minimizing the objective function. Our experiment results show that the proposed algorithm achieves throughput fairness, reduces end-to-end delay, and outperforms other general schemes and algorithms by at least 10.19%.

key words: *delay, fairness, mixed-integer nonlinear programming, performance, wireless networks*

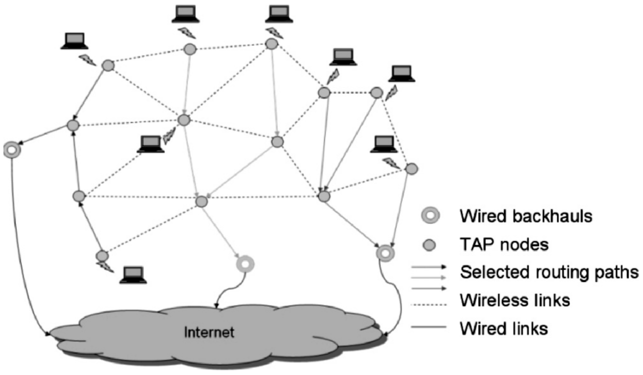


Fig. 1 A mesh network constructed with a BS-oriented structure connecting MDs to a TAP, and an ad hoc structure connecting TAPs to a wired network with backhauls. Wireless links enable this type of network to cover a large area.

1. Introduction

It is now possible to access data services anytime, anywhere via public wireless local area networks (PWLANs), which provide last-mile connectivity to the Internet at low cost. By using a mesh network, on the last-hop, such as those used in Wi-Fi wireless Internet hotspots, the major deployment and maintenance costs of wired infrastructures can be reduced, thereby reducing overall ISP costs [3].

In a multi-hop wireless mesh network (WMN), wireless nodes access the Internet via backhaul nodes (i.e., incoming or outgoing information gateways) that connect to the Internet via wired lines [10]. As shown in Fig. 1, WMNs consist of several TAPs (Transient Access Points) [11] and at least one backhaul facility. The TAPs are responsible for accepting connection requests from clients and for relaying data traffic between TAPs. In the WMN architecture, a backhaul node receives client data traffic by wireless multi-hop communications through interconnected TAPs. The traffic is then transmitted over the Internet from the gateway node.

The development of multi-hop communications in WMNs represents a milestone in wireless communications. However, some inherent characteristics of WMNs give rise to certain problems [3], [5], [6], [13], of which one of the most important is the issue of fairness [7], [14], [15]. In a WMN, many clients use the same backhaul to access the Internet; therefore, the throughput available to each client depends on the distance between it and the backhaul [11], [14], [15]. In other words, clients on longer hop paths experience lower throughput compared to those on shorter hop paths. Thus, an important topic is the fair allocation of resources to the source node based on the end-to-end delay to the destination node.

Mobile phones and video phones require guaranteed quality of service (QoS) because they are extremely sensitive to delay. If a network is well designed and the deployment of gateway nodes is optimal, mobile devices (MDs) should be able to access multimedia applications easily because each device is allocated the appropriate amount of bandwidth such that the throughput is fair and the end-to-end delay is balanced.

A number of schemes have been developed to improve fairness and QoS in WMNs so that they can handle a variety of applications. Such approaches include: fair service per flow traffic management based on hierarchically aggregated fair queuing (HAFQ) [4], top load-balanced forest routing [15], fair end-to-end bandwidth allocation and scheduling schemes [1], [7], [14], and a nominal capacity scheme [5]. Our goal is to extend previous research in these areas.

In [1], the author proposed a model that maximizes the spatial reuse of wireless channel bands. By analyzing the relationships between contending flows, it was determined that the end-to-end throughput of multi-hop flows can be

---

Manuscript received September 6, 2006.
Manuscript revised December 21, 2006.

[†] The authors are with the Dept. of Information Management, National Taiwan University, Taiwan.
[††] The author is with the Dept. of Information Management, China University of Technology, Taiwan.
a) E-mail: d89002@im.ntu.edu.tw

DOI: 10.1093/ietcom/e90–b.5.1042

maximized by minimizing flows that are subject to bandwidth allocation constraints. In this paper, we propose a bandwidth allocation strategy that not only minimizes end-to-end delay, but also achieves load-balanced routing assignment and throughput fairness.

In [14], the authors proposed a fairness scheme that addresses the following issues:

- Temporal fairness: Within a collision area, the allocation of resources can be controlled to ensure that the channel access time is fair. However, since throughput is unfair, nodes further away from the backhaul suffer from a starvation problem [14].
- Spatial bias fairness: Since channel access time is assigned uniformly on flows, more resources must be allocated to nodes further away from the destination. Throughput does not decrease with increased hop counts, but transmissions from more distant MDs take an inordinate amount of time to reach the gateway.

The above studies focus on collision areas where links share the channel frequency, which means that only one TAP can transmit data at any time. Although the number of such collision areas can be reduced in orthogonal WMNs (e.g., the WiMax solution, which uses a different channel frequency for each directional antenna) by using an adjustable radius to reduce the over-reach interference of TAPs deployed in fixed positions, spatial bias still causes throughput anomalies. As this interference can not be removed completely, it affects the SNR (signal-to-noise ratio) value, which in turn affects in the channel capacity ($C_{(u,v)}$) on link (u, v) (u and v denote TAP nodes). To resolve this problem, we allocate an appropriate backhaul to each TAP node, and then allocate bandwidth to each link on the routing path that connects each TAP to the assigned backhaul node.

To solve the spatial bias problem, we adopt the fair queuing model with a distributed scheme used in [4]. As the implementation of bandwidth allocation issues is addressed, we focus on fair allocation of bandwidth and end-to-end delay issues. Note that, in this paper, the transmission delay is minimized because the routing path is not pre-determined.

The remainder of the paper is organized as follows. In the next section, we analyze fairness schemes and provide examples to illustrate how they resolve fairness and QoS issues. In Sect. 3, we formulate a mixed-integer nonlinear programming model that considers end-to-end delay, throughput fairness, load-balanced routing, and bandwidth assignment. In Sect. 4, we propose an algorithm that obtains good feasible solutions to the above problems. Section 5 details the experiment results. Then, in Sect. 6, we present our conclusions.

2. Problem Description

In this section, we consider bandwidth allocation schemes that minimize end-to-end delay, achieve load-balanced routing assignment, and balance the delay and throughput.

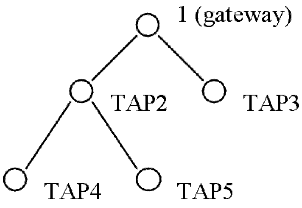


Fig. 2 Tree-based fairness analysis of a WMN topology.

2.1 Fair End-to-End Delay

Fairness issues in WMNs focus on requiring traffic to follow a uniform distribution so that the assignment of aggregated flows between MDs and TAPs is balanced. Figure 2 shows a simple WMN topology in which the link capacity $C_{(u,v)} = 1$ (where u and v denote TAP nodes).

- Temporal fairness: We consider two temporal fairness issues in networks where no collisions occur in the transmission area: (i) how to ensure that the transmission time for each link is fair; and (ii) how to ensure that the end-to-end transmission time is fair. In the first case, each link has the same amount of time to transmit packets to a gateway; therefore, the cycle time (i.e., the time, usually measured in microseconds or nanoseconds, between the start of one transmission to the time when the next transmission can be started) increases when there is more flow on one link compared to other links, as shown by CASE I in Table 1. For example, the cycle time of link $(2, 1)$ is $3(= 3/1)$ when TAP nodes 2, 4, and 5 have 1 unit of data per unit of time and the link bandwidth is 1 (i.e., $c_{s(u,v)} = 1$). As a result, the maximum end-to-end delay is 4 (= 1 on link (4,2) plus 3 on link (2,1) for TAP4) and the total throughput is 4 when the traffic requirement for each node is 1. In Table 1, d denotes the maximum end-to-end delay; $t_{s(u,v)}$ denotes the transmission time per unit of data for TAP node s on link (u, v); ρ denotes the total throughput; $c_{s(u,v)}$ denotes the amount of bandwidth allocated to TAP node s on link (u, v); and ρ denotes the total throughput.
  In the second case, each link is allocated the same amount of bandwidth, but each flow takes only a single unit of time per cycle for end-to-end transmission. CASE II in Table 1 shows the allocation results with maximum throughput equal to 2.5 (= 1 by TAP2, 1 by TAP3, 0.25 by TAP4, and 0.25 by TAP5) when the maximum end-to-end delay time $d = 3$ (= 0.5 on link (4,2) plus 2 on link (2,1) for TAP4). These results are the similar to those reports [14], which show that TAPs further away from the gateway require more hops to transmit data and more flow to transmit on a link than those closer to the gateway. Thus, the maximum end-to-end delay is longer and the throughput is uneven.
- Spatial bias fairness: As noted in Sect. 1, spatial bias is a major concern in WMNs. To be fair, the end-to-end

Table 1 Analysis of different fairness schemes in WMNs.

CASE I: Temporal fairness with fair transmission time on each link.							
	TAP2	TAP3	TAP4		TAP5		
link	(2,1)	(3,1)	(4,2)	(2,1)	(5,2)	(2,1)	
$t_{s(u,v)}$	1	1	1	1	1	1	$d = 4$
$c_{s(u,v)}$	1	1	1	1	1	1	
Throughput	1	1	1		1		$\rho = 4$
CASE II: Temporal fairness with fair end-to-end transmission time.							
	TAP2	TAP3	TAP4		TAP5		
link	(2,1)	(3,1)	(4,2)	(2,1)	(5,2)	(2,1)	
$t_{s(u,v)}$	1	1	0.5	0.5	0.5	0.5	$d = 3$
$c_{s(u,v)}$	1	1	1	1	1	1	
Throughput	1	1	0.25		0.25		$\rho = 2.5$
CASE III: Non-spatial bias fairness with fair throughput.							
	TAP2	TAP3	TAP4		TAP5		
link	(2,1)	(3,1)	(4,2)	(2,1)	(5,2)	(2,1)	
$t_{s(u,v)}$	1	1	1	1	1	1	$d = 4$
$c_{s(u,v)}$	1	1	1	1	1	1	
Throughput	1	1	1		1		$\rho = 4$
CASE IV: Fair end-to-end transmission time with equal throughput.							
	TAP2	TAP3	TAP4		TAP5		
link	(2,1)	(3,1)	(4,2)	(2,1)	(5,2)	(2,1)	
$t_{s(u,v)}$	5	5	2.5	2.5	2.5	5	$d = 5$
$c_{s(u,v)}$	0.2	0.2	0.4	0.4	0.4	0.4	
Throughput	1	1	1		1		$\rho = 4$
CASE V: Fair end-to-end transmission time on each link with equal throughput. The remaining resources are also allocated.							
	TAP2	TAP3	TAP4		TAP5		
link	(2,1)	(3,1)	(4,2)	(2,1)	(5,2)	(2,1)	
$t_{s(u,v)}$	5	5	5	5	5	5	$d = 5$
$c_{s(u,v)}$	0.2	1	0.4	0.4	0.4	0.4	
Throughput	1	5	1		1		$\rho = 8$
CASE VI: Fair end-to-end transmission time and throughput with variable bandwidth allocation.							
	TAP2	TAP3	TAP4		TAP5		
link	(2,1)	(3,1)	(4,2)	(2,1)	(5,2)	(2,1)	
$t_{s(u,v)}$	3.731	3.731	1	2.731	1	2.731	$d = 3.731$
$c_{s(u,v)}$	0.268	0.268	1	0.366	1	0.366	
Throughput	1	1	1		1		$\rho = 4$
CASE VII: Fair end-to-end transmission time and throughput with variable bandwidth allocation. The remaining resources are also allocated.							
	TAP2	TAP3	TAP4		TAP5		
link	(2,1)	(3,1)	(4,2)	(2,1)	(5,2)	(2,1)	
$t_{s(u,v)}$	3.731	3.731	1	2.731	1	2.731	$d = 3.731$
$c_{s(u,v)}$	0.268	1	1	0.366	1	0.366	
Throughput	1	3.731	1		1		$\rho = 6.731$

throughput for each subscriber should be equal. However, as CASE III in Table 1 demonstrates, more distant MDs take longer to achieve the same throughput as TAPs that are close to a gateway. Clearly, transmission times are unfair under this scheme, even in situations where the system throughput is equal to 2.5 (1 by TAP2, 1 by TAP3, 0.25 by TAP4, and 0.25 by TAP5) but the maximum end-to-end delay time $d = 4(e.g., t_{4(4,2)} = 1/1, t_{4(2,1)} = 1/1$. However, the bandwidth allocated to TAP2 is 1 and the amount allocated to TAP5 is also 1, so the cycle time on link (2,1) is 3. Therefore, for TAP4, $d = 1+3 = 4$). In this case, nodes further away, i.e., TAP4 and TAP5, take twice the time taken by TAP2 or TAP3 to balance the throughput.

- Delay fairness: By allocating the same amount of bandwidth to each link, the above schemes control the transmission time or throughput to achieve temporal fairness and avoid spatial bias. In contrast, we allocate variable bandwidth by a delay fairness scheme, such that both throughput and end-to-end delay are fair simultaneously. Thus, links with the same amount of flow are allocated equal bandwidth. CASE IV in Table 1 shows that the bandwidth for TAP4 is allocated to the next two links (i.e., (4,2) and (2,1)), each of which has 0.4 units of bandwidth; however, the bandwidth is only 0.2 units on link (2,1) for TAP2 (i.e., the bandwidth 1 on the critical link (2,1) is fully allocated to TAP2 ($c_{2(2,1)} = 0.2$), TAP4 ($c_{4(2,1)} = 0.4$), and TAP5 ($c_{5(2,1)} = 0.4$)). Accordingly, the maximum delay time $d = 5$ (derived from $t_{4(4,2)} = 1/0.4$, $t_{4(2,1)} = 1/0.4$, and $d = t_{4(4,2)} + t_{4(2,1)} = 5$ for TAP4) and the throughput of each TAP is 1.

  In CASE IV, the allocation of resources is restricted by a bottleneck link (i.e., link (2,1)). In this case, other flows (e.g., link (3,1)) can utilize the remaining bandwidth, thereby increasing system throughput. CASE V in Table 1 shows the bandwidth allocation when the bandwidth of TAP3 is changed to 1, as per the min-max concept. Even though the total throughout is 8 units per unit of time (1 unit for TAP2, 5 for TAP3, 1 for TAP4, and 1 for TAP5), it does not improve other critical TAP nodes.
- Extended throughput and end-to-end delay fairness: Extended throughput and end-to-end delay fairness are controlled by varying the bandwidth assigned to flows. According to the principle of variable bandwidth, network bandwidth is allocated optimally along critical paths, as shown by CASE VI in Table 1. The bandwidth for TAP4 is allocated to the following two gateway links: (i) on link (4,2), the bandwidth is 1, so transmitting one unit of data takes one unit of time; and (ii) on link (2,1), the bandwidth is 0.366 (derived by simultaneous functions $1 + t_{4(2,1)} = 1 + t_{5(2,1)} = t_{2(2,1)}$ and $t_{4(4,2)} = 1/c_{4(2,1)}$), so transmitting one unit of data takes 2.731 units of time. The single hop flow from TAP2 to the gateway is allocated a total bandwidth of 0.268, so transmitting one unit of data takes 3.731 (i.e., $t_{4(4,2)} = 1/1$, $t_{4(2,1)} = 1/0.386$, and $d = t_{4(4,2)} + t_{4(2,1)} = 3.731$ for TAP4) units of time.

  Under this scheme, end-to-end delay and throughput are equal for all TAPs. The maximum end-to-end delay, d, is 3.731. Using the same allocation scheme as that in CASE VI, but optimizing the bandwidth allocation for paths without bottlenecks, as in Case IV, improves the total system throughput. CASE VII shows that the entire bandwidth on the link (3,1) is allocated to TAP3. The throughput of TAP3 is 3.731 and the system throughput is 6.731.

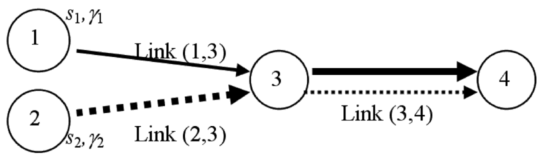


Fig. 3 Capacity assignment control for achieving fair end-to-end delay.

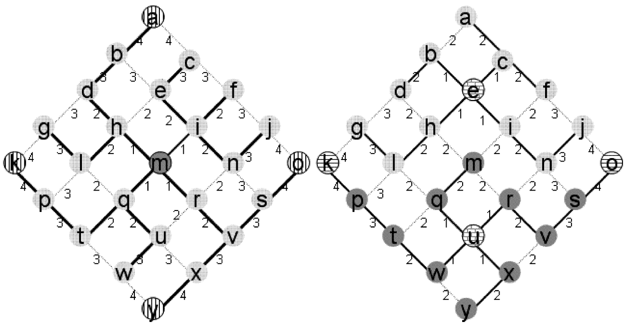
2.2 Fair Bandwidth Allocation

To summarize the above analysis, CASE VII not only resolves the fairness issue, but also optimizes the allocation of resources to solve the spatial bias problem. **Lemma 1** defines the two conditions required to achieve our objective. Accordingly, to minimize the maximum end-to-end delay, we adopt the variable bandwidth allocation and fairness strategy in CASE VII, and model the problem as a mathematical formulation. In complex networks, which have longer paths and increased flow on the links, end-to-end delay increases; therefore, the total throughput is lower. This gives us a hint about devising a routing algorithm, namely, the solution to the problem is provided by the shortest disjoint path routing algorithm, which we discuss in Sect. 4.

Lemma 1: The two conditions necessary to achieve min-max end-to-end delay are: (i) the bandwidth must be assigned completely; and (ii) the capacity must be fairly allocated among the links.

Proof: Assume that the traffic requirement γ_s of all nodes is the same or weighted and the link bandwidth of each link is constant $C_{(u,v)}$. If $\sum_{s \in V} c_{s(u,v)} < C_{(u,v)}$, the min-max delay, d, is reduced as the remaining capacity is allocated. Thus, a necessary condition for achieving minimal end-to-end delay is that the bandwidth must be completely allocated. Assume that the minimum end-to-end delay is equal to $\overline{d}$ when the entire bandwidth is fairly and completely allocated to all flows s. If all flows on the path links are allocated the same capacity $\widehat{c}_{s(u,v)}$, except one link (u', v') where flow s' is assigned less bandwidth $c_{s'(u',v')}$ (i.e., $c_{s'(u',v')} < \widehat{c}_{s(u,v)}$), the end-to-end delay of flow s' will be longer. The reason is that the end-to-end delay of s' is $d_s = \sum_{(u,v) \in (L_s \backslash (u',v'))} D_{(u,v)}(c_{s(u,v)}, \gamma_s) + D_{(u,v)}(c_{s(u',v')}, \gamma_s)$ where $D_{(u,v)}$(the bandwidth,traffic requirement rate) is the function that calculates the delay time. Then, $d_{s'} > d_s$ because $c_{s'(u',v')} < c_{s(u',v')}$. Now, the objective delay value d is equal to the delay of flow s' and $\overline{d} < d_{s'}$ which increases the end-to-end delay.

In this situation, if the capacity allocated to flow s' is higher than that for any other flow s on link (u'', v'') (i.e., $c_{s'(u'',v'')} > c_{s(u'',v'')}$), the end-to-end delay will be balanced because the end-to-end delay of flow s' is $d_{s'} = \sum_{(u,v) \in (L_s \backslash (u',v') \backslash (u'',v''))} \begin{pmatrix} D_{(u,v)}(c_{s(u,v)}, \gamma_s) + \\ D_{(u',v')}(c_{s'(u',v')}, \gamma_s) + \\ D_{(u'',v'')}(c_{s'(u'',v'')}, \gamma_s) \end{pmatrix}$, and the other flows are $d_s = \sum_{(u,v) \in L_s} D_{(u,v)}(c_{s(u,v)}, \gamma_s)$, $\forall s \in V \backslash s'$.Then, $d_{s'}$ will be equal to d_s if the bandwidth allocation scheme is appropriate.



(a) TLBT topology (b) TLBF topology

Fig. 4 (a) One backhaul (node *m*) is assigned in a TLBT topology; and (b) two backhauls (nodes *e* and *u*) are assigned in a TLBF topology.

Therefore, the objective value d is equal to $d_{s'}$ or d_s, but $d_{s'} > \overline{d}$ due to the mean value inequality. In summary, to achieve min-max end-to-end delay, the bandwidth must be fairly allocated to each link based on the traffic requirement.

2.3 Load-Balanced Routing

In addressing the routing issue, we focus on: (i) the creation of "load-balanced forests" in which network traffic flows evenly; and (ii) the appropriate assignment of backhauls. Load-balanced routing maximizes the utilization of resources and reduces end-to-end delay.

Some studies, such as [2] and [9], use a single backhaul to balance the traffic load on the incoming link of an egress node, which is considered the primary bottleneck. The purpose is to determine a top-level load-balanced tree (TLBT) topology, as shown in Fig. 4(a). A TLBT topology can be applied to networks with multiple sources because they route to a single destination. A balanced tree is constructed with three heuristics, namely: the best first, random, and weighted heuristics. The heuristics use a weight function ($s'(t_A) = m \cdot f_A +$ the number of nodes in t_A), where m is a weight, f_A denotes the aggregated flow for node A, and t_A denotes the sub-tree of node A. The function that generates the fairness index [12] is calculated by:

$$\beta(b) = \left(\sum_{i=1}^{k} f_{X_i}\right)^2 \Big/ k \sum_{i=1}^{k} f_{l_i}^2$$

where b denotes an egress node, k denotes the number of adjacent links, and f_{l_i} denotes the aggregated traffic load on an adjacent link i.

A mesh network constructed with several backhaul nodes, denoted as tree roots or cluster heads in a graph, is called a "forest." We adopt a TLBT topology and extend it to a "top-level load-balanced forest" (TLBF) [15], as shown in Fig. 4(b). A TLBF is balanced as a backbone forest for a set of variant loads such that all backhauls and all branches, which are the closest links to the backhauls, carry approximately equal amounts of traffic. As shown in Fig. 4(b), the TLBF can use different methods (i.e., a greedy algorithm, a mathematical model, and a heuristic approach) to support load-balanced multiple backhauls in a WMN.

3. Problem Formulation

The findings reported in Sect. 2 are used to facilitate inter-TAP temporal and throughput fairness as well as load-balanced routing and bandwidth allocation. The problem is modeled as a graph of connected TAP nodes, $\Gamma(V, L)$, where V represents the set of nodes (i.e., TAPs) distributed over an area; and $(u, v) \in L$ denotes the direct links such that node v can receive signals from node u via an orthogonal channel. The set of candidate backhauls, B, is also given. The problem description is summarized right column.

The problem is formulated as follows: How to optimize the objective function (IP) based on the expression of the min-max end-to-end transmission time, denoted by d,

$$Z_{IP} = \min d \tag{IP}$$

Given:

- · The set of TAPs V.
- · The set of backhauls B.
- · The set of candidate paths by which a TAP reaches a backhaul P_{bs}.
- · The set of links L.
- · Indication variable $\delta_{p(u,v)}$, which denotes link (u, v) on the path p.
- · The given link capacity $C_{(u,v)}$, where $(u, v) \in L$.
- · The traffic requirement for each TAP, modeled as a Poisson distribution with rate γ_s (unit/sec).

Object:

To minimize the maximal end-to-end delay of a WMN.

Subject to the following constraints:

- · Backhaul selection: each TAP must select a backhaul as its gateway.
- · Routing: one path to a backhaul must be found in order to transmit and receive data.
- · Link: each TAP selects a backhaul as its gateway. The selected links of all routing paths to the node form a forest.
- · Capacity: the aggregated flow of each link is limited by the capacity constraint.
- · Delay: the delay constraint calculates the maximum end-to-end delay that can be minimized by the objective function.

To determine:

- · The backhaul a TAP should select to transmit data.
- · The routing path from the TAP to the backhaul.
- · Whether a link should be selected for the routing path.
- · A top-level load-balanced mesh network.
- · The TAPs that should be selected to form the set of backhauls.
- · The capacity allocated to the selected links of a TAP.
- · The node-to-node delay on the selected links of a TAP; and
- · The maximum end-to-end delay of the WMN

---

subject to the following constraints:

A. Backhaul selection constraints

The decision variable $z_{bs} \in 0, 1$ is used to indicate whether a TAP routes to a backhaul b. Since each TAP s has to route to a single backhaul b, we can formulate our first constraint. For each TAP, Constraint (1) limits the summation of all selected backhauls for each TAP to 1.

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

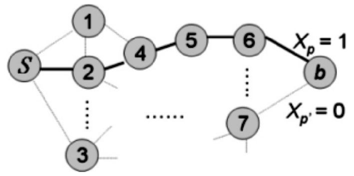


Fig. 5 The paths from s to b are denoted by the set P_{bs}. The upper path (bold line) is the only active path selected from this set, so $x_p = 1$ denotes that this is the selected path p $(s, 2, 4, 5, 6, b)$. For all other paths p' (e.g., path $s, 3, \ldots, 7, b$), $x_{p'} = 0$, as no other paths are selected.

B. Path constraints

Whether a path $p \in P_{bs}$ is used to connect a TAP s to a backhaul b depends on the value of the decision variable x_p. If $x_p = 1$, the path will be used; otherwise, it will not be used. To ensure that each TAP only finds one path to its backhaul, Constraint (2) requires that:

$$\sum_{b \in B} \sum_{p \in P_{bs}} x_p = 1, \ \forall s \in V \tag{2}$$

The decision variables x_p and z_{bs} are related such that, once a path p has been selected, the decision variable z_{bs} must be set to 1. This constraint is described by (3). Note that the limits on the right-hand side of (3) mean that only one path from P_{bs} can be selected.

$$\sum_{b \in B} z_{bs} \leq \sum_{b \in B} \sum_{p \in P_{bs}} x_p, \ \forall s \in V \tag{3}$$

Figure 5 shows a scenario where one path is selected from the set of candidate paths P_{bs} to connect TAP s to any backhaul b once node s has selected backhaul b as its backhaul (i.e., $z_{bs} = 1$). Then, x_p is set to 1, and the other path, p', is set to 0 (i.e., $x_{p'} = 0$). Since this is a shortest path problem, the path p (i.e., $x_p = 1$) can be found by existing algorithms (e.g., Dijkstra's algorithm). Thus, the set of paths P_{bs} does not need be listed.

C. Link constraints

In Constraint (4), the decision variable $y_{(u,v)} = 1$ indicates that a link $(u, v) \in L$ is used (e.g., in Fig. 5, $y_{(2,4)} = 1$ because link (2,4) is used); and $y_{(u,v)} = 0$ indicates that a link (u, v) is not used (e.g., in Fig. 5, $y_{(2,1)} = 0$ because link (2,1) is not used).

$$y_{(u,v)} = 0 \text{ or } 1, \ \forall (u, v) \in L \tag{4}$$

Thus, the relationship between path x_p and link decision variable $y_{(u,v)}$ is such that once a path p from TAP s to any candidate backhaul has been selected and a link (u, v) is on that path, the decision variable $y_{(u,v)}$ must be set to 1. For example, in Fig. 5, path p is selected, as shown by the bold line. The decision variable $y_{(u,v)}$ for the following links $(s, 2)$, (2,4), (4,5), (5,6), and $(6, b)$ is set to 1; otherwise it remains at 0. This constraint is described by (5):

$$\sum_{b \in B} \sum_{p \in P_{bs}} x_p \delta_{p(u,v)} \leq y_{(u,v)}, \ \forall s, u, v \in V \tag{5}$$

where $\delta_{p(u,v)}$ is the indicator function. Note that the function is equal to 1 when a link (u, v) is on a path p, where $p \in P_{sb}$, $(u, v) \in L$; otherwise, it is equal to 0.

The problem (IP) can be viewed as the construction of multiple trees whose roots are given by the set of candidate backhauls. Three link constraints, (6)–(8), restrict the structure of the trees, thereby ensuring that all data can be transmitted via a path from gateways to TAP nodes or vice versa.

- The number of out-degree links for each TAP node, except a gateway node, can not be more than 1, as shown in (6):

$$\sum_{v \in V} y_{(u,v)} \leq 1, \ \forall u \in V \tag{6}$$

- At least one link must connect to b when at least one TAP selects b as its backhaul (i.e., $z_{bs} = 1$, $s \in S$), so the summation of in-degree links for the backhaul is at least 1, as shown in (7):

$$\sum_{v \in V} y_{(v,b)} \geq z_{bs}, \ \forall b \in B; s \in V \tag{7}$$

- Based on (i) and (ii), the total number of selected links will be equal to the total number of TAPs minus the total number of selected backhauls. Thus, Constraint (8) is added to restrict the number of selected links:

$$\sum_{u \in V} \sum_{v \in V} y_{(u,v)} \geq |V| - |B| \tag{8}$$

D. Capacity constraint

The capacity constraint (9) ensures that the aggregated flow (i.e., the bandwidth allocated to each link) does not exceed the given capacity $C_{(u,v)}$ of a link (u, v).

$$\sum_{s \in V} c_{s(u,v)} \leq C_{(u,v)}, \ \forall u, v \in V \tag{9}$$

The allocation of bandwidth must fulfill two conditions: (i) the bandwidth must not exceed the link capacity; and (ii) bandwidth is only allocated to a link (u, v) if that link is used. This constraint is given by (10):

$$c_{s(u,v)} \leq C_{(u,v)} \sum_{b \in B} \sum_{p \in P_{bs}} x_p \delta_{p(u,v)}, \ \forall s, u, v \in V \tag{10}$$

E. Delay constraints

The transmission time for data sent from a TAP s to a backhaul via a link (u, v) can be calculated according to the amount of bandwidth allocated, as shown by Eq. (11). The value of the left-hand side of the equation is only computed when the link (u, v) is used; otherwise, it is equal to 0. The amount of data on TAP s, denoted by s, which must be transmitted, is given. When the link is used, the bandwidth, denoted by the decision variable $c_{s(u,v)}$ is allocated to TAP s on the link (u, v). Then, the node-to-node delay time for the link (u, v) on the routing path of node s is calculated as the variable $t_{s(u,v)}$ on the right-hand side of the equation. Note that an arbitrary small number (where $\varepsilon < 10^{-6}$) is added to avoid mathematical errors caused by $c_{s(u,v)} = 0$. This small value does not affect the solution when only reasonable delays occur on the selected links with $\sum_{b \in B} \sum_{p \in P_{bs}} x_p \delta_{p(u,v)}$ values.

$$\frac{\gamma_s \sum\limits_{b \in B} \sum\limits_{p \in P_{bs}} x_p \delta_{p(u,v)}}{c_{s(u,v)} + \varepsilon} = t_{s(u,v)}, \ \forall s, u, v \in V \tag{11}$$

Constraint (12) is used to find the maximum end-to-end transmission time (i.e., end-to-end delay) d by summing the transmission times of the links on the paths in V. As the objective function is to minimize d, the problem is formulated as a min-max problem. Allocation of the resources is optimized for non-critical paths under the capacity constraint (9).

$$\sum_{u,v \in V} t_{s(u,v)} \leq d, \ \forall s \in V \tag{12}$$

4. Proposed Algorithms

We apply the min-max concept to achieve both minimum end-to-end delay and to resolve the issue of load-balanced routing. Although this is an optimization-based problem, which generally calls for a centralized mechanism, its solution may be determined by a distributed mechanism. Thus, we propose a solution for balancing the end-to-end delay of a sub-tree.

4.1 Load-Balanced Routing Algorithm

Recall that when paths are longer and there is more traffic on the links, throughput is lower and end-to-end delay is longer. As a result, determining the shortest and least-used paths to the backhaul node provides a good hint for devising the proposed algorithms. To solve this routing problem, we previously proposed a greedy-based approach called greedy load-balanced routing (GLBR) [15]. Here, we extend that algorithm to support the delay issue; in other words, the link cost is calculated by the delay function $D_{(u,v)}(C_{(u,v)}, \gamma_u)$.

To satisfy Constraints (4) to (8), one link on a path associated with a given backhaul is selected per iteration to connect the TAP to its next hop. Initially, all link costs are set to $\zeta_{(u,v)} = D_{(u,v)}(C_{(u,v)}, \gamma_u)$ and all TAPs, except the given backhauls (i.e., $V - B$), are marked as FALSE. We then assign the link with the lowest cost to the dominant forest set T as a function of the number of out-degrees of the current total traffic flow at the egress link. Once a link on the routing path has been selected, the cost of the out-degree is equal to aggregated cost at the previous node plus $\zeta_{(u,v)}$ so that the traffic flow of each branch is balanced. The aggregated cost of a node s that belongs to the selected path is also increased by $\zeta_{(s,v)}$. The above steps are repeated until all nodes have been assigned to the forest dominant set T so that Constraints (1) to (3) are satisfied.

Algorithm 1 shows the pseudo code for the GLBR algorithm, which extends the concept of Prim's minimum cost spanning tree algorithm to achieve the minimum flow of a

TLBF in a WMN. The "while-loop," beginning on line 8, ensures that each node not yet assigned to a backhaul is selected to connect to one backhaul to fulfill Constraints (2) and (3). Lines 10 restricts the selection of a link to one previous node for each TAP, thereby fulfilling Constraints (5), (6), and (9). Lines 14–16 and 17–19 increase the link cost and capacity per iteration, respectively, to meet the min-max concept (load-balance requirement) of the objective function (IP) and Constraint (12).

The time complexity of the load-balanced routing algorithm is $O(V^2)$, determined by finding a node to associate to a backhaul in every loop.

Algorithm 1 GLBR (B, V, L)

```
Require: $\Gamma = (V, L)$ (a weighted directed graph, where $v \in V$ and link $(u, v) \in L$), and a set of backhauls $B$.
Ensure: The nodes of the routing tree are included in the dominant forest set $T$. The variable, $v.pred$ records which previous node of each relay node $v$ associates to a backhaul. The variable $\zeta_{(u,v)}$ denotes the link cost $(u, v)$, which is calculated by the delay function.
 1: for all link $(u, v)$ do
 2:     $\zeta_{(u,v)} = (D_{(u,v)}(C_{(u,v)}, \gamma_u)$
 3: end for
 4: for all backhaul $b$ do
 5:     $T = T \cup b$;
 6:     $b.pred = NULL$;
 7: end for
 8: while node $v$ is not in $T$ do
 9:     for all node $u$ in $T$ do
10:         FIND the minimum cost node $v$ that is not included in the dominant set $T$. Here, we also check the capacity constraint (9) along the path $p$. If more than two links $(u, v)$ have the same cost, we select the minimal branch cost of node $u$ and the minimal number of out degrees of node $v$.
11:     end for
12:     $T = T \cup v$;
13:     $v.pred = u$;
14:     for all node $s$, which is not in $T$, have the direct link to $v$ do
15:         LET link cost $(s, v)$ = link cost $(s, v) + \zeta(s, v)$;
16:     end for
17:     while $v.pred$ is not a backhaul do
18:         INCREASE the link cost to $\zeta_{(u,v)}$ along the selected path from node $v$.
19:     end while
20: end while
```

4.2 Fair Bandwidth Allocation Algorithm

Once the routing path has been determined, the bandwidth allocation for each selected link of a TAP is calculated independently along the visited links and relayed to the gateway node by the DFS (Depth First Search) algorithm to make decisions about (9) and (10). The node-to-node delay $t_{s(u,v)}$ and maximum end-to-end delay d are then calculated by (11) and (12), respectively.

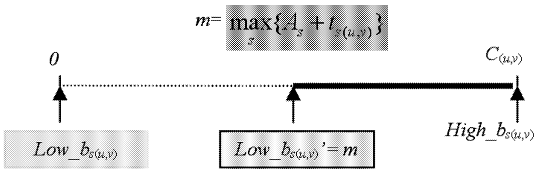
Algorithm 2 shows a variable bandwidth allocation algorithm called EDTB, which establishes fair end-to-end delay and throughput. The path from each node to its gateway node, determined by the GLBR algorithm, is traced once, and visited links are recorded for each TAP node s that links (u, v) in the set $t_{s(u,v)}$. According to the objective function, finding the min-max end-to-end delay balances the delay of all pairs of nodes from the TAP node s to the backhaul b. In this fashion, the bandwidth allocation scheme is executed on all links using a modified binary-search method based on the "Resource Exchange" property of the fairness condition [12].

Algorithm 2 EDTB (V, L, T)

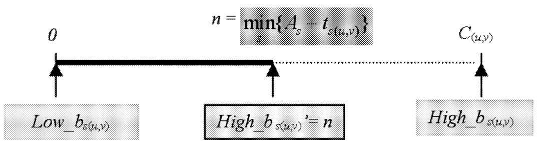
```
Require: Routing paths included in the dominant set $T$ with a variable, $v.pred$, which marks the previous node of each relay node $v$ to the backhaul $b$.
Ensure: The bandwidth and transmission time allocated to each node $s$ on the link $(u, v)$ selected for the $p \in P_{bs}$.
for all vertices $v$ do
    if the node $s$ is a leaf node then
        ALLOCATE bandwidth $b_{s(u,v)}$ to the outgoing link of node $s$. The link transmission time $t_{s(u,v)}$ and aggregated transmission time $A_s$ of source node $s$ are calculated.
    end if
    while v.pred != NULL do
        Link set $l \cup (u, v)$ {Link set $l$ records the selected link for each source node $s$.}
        $n_s$ = the number of paths that use the link $(u, v)$;
    end while
end for
$\psi \cup B$ {where nodes are visited}
for all node not in $\psi$ do
    FIND a node $u$ that all incoming nodes $s$ have visited.
    SET $Low[s] = 0$ and $High[s] = C_{(u,v)}$ for all nodes $s$.
    ALLOCATE the equivalent amount of resources $r = C_{(u,v)}/n_s$ to node $s$.
    repeat
        CALCULATE the aggregated delay time $A_s$
        $m = \arg\max_{s \in V}\{A_s + t_{s(u,v)}\}$; {link $(u, v)$ is in link set $l$}
        $n = \arg\min_{s \in V}\{A_s + t_{s(u,v)}\}$; {link $(u, v)$ is in link set $l$}
        $Low[m] = r$;
        $High[n] = r$;
        $r = (High[m] - Low[m])/n_s$;
        MOVE $r$ bandwidth from $n$ to $m$;
    until $r < \varepsilon$
    UPDATE the aggregation time $A_s$;
end for
OUTPUT: resource allocation results and transmission time for each link of each node $v$;
```

The time complexity of the EDTB algorithm is $O((V + L)I)$, determined by the sequential trace node of the DFS concept. Here, the variable I indicates the number of iterations required per node to adjust the bandwidth per node.

Figure 6 shows the aggregated transmission time at the source node of a sub-tree plus the current transmission time. When this value is maximal, the bandwidth allocated to the path will be greater than the average bandwidth, so the transmission times will be reduced, as shown in Fig. 6(a). Conversely, when the value is minimal, the range of allocated bandwidth should be less than the current amount of allocated bandwidth, as shown in Fig. 6(b). In the latter situation, the surplus allocated bandwidth is shifted from the shortest transmission path to the longest transmission path. Thus, transmission times will be almost equal and all nodes

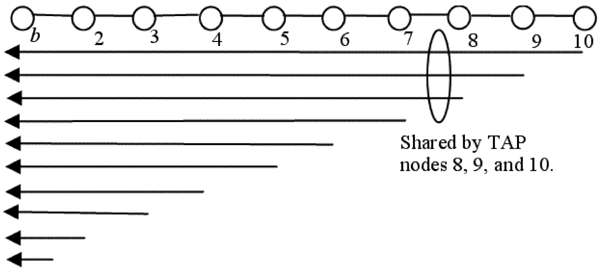


(a) Since $A_s + t_{s(u,v)}$ is the maximum, which means node s has less resources, the node's bandwidth must be increased to reduce the transmission time. Thus, we let $Low\_b_{s(u,v)}$ be equal to m so that the search area is limited to the right half of the search range.



(b) Since $A_s + t_{s(u,v)}$ is the minimum, which means node s has the most resources, the node's bandwidth must be reduced to give other nodes more bandwidth. Thus, we let $High\_b_{s(u,v)}$ be equal to n so that the search area is limited to the left half of the search range.

Fig. 6 The requirement of fair channel access time based on a modified binary search is fulfilled because the "Resource Exchange" property [12] of the fairness condition is satisfied.



Links	2-b	3-2	4-3	5-4	6-5	7-6	8-7	9-8	10-9
TAP2	0.025								
TAP3	0.122	0.032							
TAP4	0.122	0.138	0.041						
TAP5	0.122	0.139	0.158	0.055					
TAP6	0.122	0.138	0.161	0.189	0.078				
TAP7	0.122	0.138	0.160	0.189	0.231	0.118			
TAP8	0.122	0.138	0.160	0.189	0.231	0.294	0.196		
TAP9	0.122	0.138	0.160	0.189	0.231	0.294	0.402	0.382	
TAP10	0.122	0.138	0.160	0.189	0.231	0.294	0.402	0.618	1.000

Fig. 7 An example of bandwidth allocation for a chain, where the bandwidth is fairly allocated iteration-by-iteration starting from the leaf node. When the total throughput of each TAP is 1 and the capacity of each link is 1, the maximum end-to-end delay $d = 39.81$.

will have the same throughput.

Figure 7 shows the experiment results for a simple situation where the traffic requirement of each node is 1 and the bandwidth is allocated from leaf node 10 to the backhaul node b. The full bandwidth is only assigned to a TAP node once, i.e., when the node (node 10) forms a link (10,9). The node is then marked as 'visited' and all subsequent incoming nodes must also be visited. In this case, node 9 was the previous node traced, so the aggregated time spent in the sub-tree must be considered on this link. Thus, the end-to-end delay and throughput are fair after any intermediate node forwards their previous nodes' and its data. The amount of bandwidth allocated to the previous nodes is the same. Only the current node is allocated less bandwidth to order to balance the transmission time.

The results in Fig. 7 exhibit the following interesting properties: (i) the bandwidth for each incoming flow can be allocated to each node independently; (ii) the total proportion of allocated bandwidth is one (utilization); and (iii) the proportion of flow from each of the previous nodes (not required by this node) is the same. A necessary condition for (i) is that the traffic requirement routing via a node must be known. This means the available bandwidth on each link can be fully shared as, according to (ii), the total proportion is one. To reduce the complexity of the EDTB algorithm, the results can be calculated easily by (11) (i.e., only the current flow and aggregated flow are considered.)

Our proposed routing algorithm, an extension of Prim's algorithm that visits one node rooted with each candidate backhaul per iteration. It can be applied to different sizes and shapes of network. Since each node has only one outgoing link, the bandwidth allocation algorithm is executed when its subtree is allocated. The trace sequence is based on DFS, which is a converged algorithm.

5. Evaluation and Experiment Results

For evaluation purposes, the mesh network used in the experiments was comprised of N TAP nodes distributed over an area. Some nodes were assigned as backhauls (i.e., $b \in B$). The following four conditions were compared: (i) variations in the number of hops in a chain, as shown in Fig. 8; (ii) variations in the number of nodes; (iii) variations in the number of backhauls; and (iv) variations in the number of nodes to compare different routing algorithms with random topologies. When the number of nodes deployed in a fixed area is increased randomly, the traffic load and density also increase. The number of hops of between some nodes also increases. Our objective is to evaluate how the proposed algorithms reflect the traffic load and node density. We also devised a "fair bandwidth allocation per node" scheme to compare (i)–(iii).

All the experiments were performed on a PC with a 1.3 GHz CPU and 1.0 GB of DRAM. The operating system was Linux Red Hat 9.0 with kernel version 2.4.20, and the code was written using the C programming language, compiled by GNU gcc version 3.2.2. The execution time was approximately 3.1 seconds for 250 nodes with $\varepsilon < 10^{-6}$.

Figure 8 shows the experiment results for condition (i). The proposed advanced delay fairness algorithm, EDTB, achieves the lowest end-to-end delay, compared to the spatial bias fairness scheme, fair bandwidth allocation per node scheme, and temporal fairness scheme by 11.5–15.0%, 51–60%, and 34.5–50.8%, respectively (the routing algorithm is GLBR). In a WMN, the maximum end-to-end delay is significantly lower than under other schemes, especially when the number of hops increases. This is because the proposed algorithm allocates the complete bandwidth, link-by-link, on a path with a bottleneck.

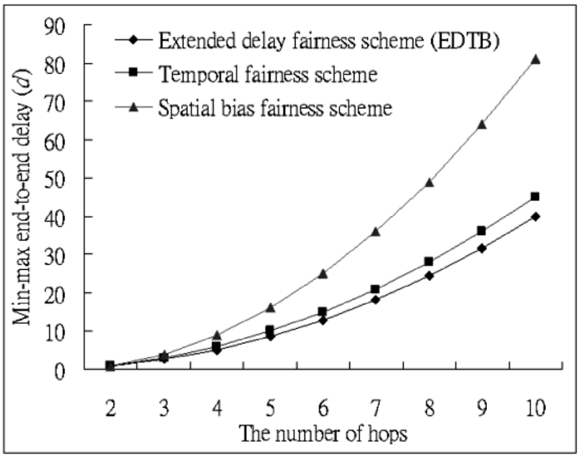


Fig. 8 Experiment results using different numbers of hops.

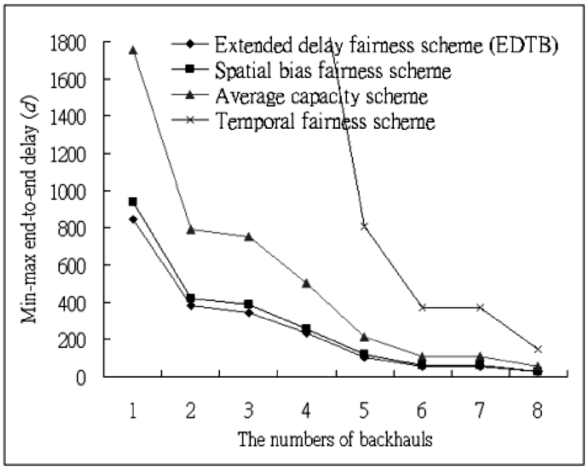


Fig. 10 Experiment results using different numbers of backhauls.

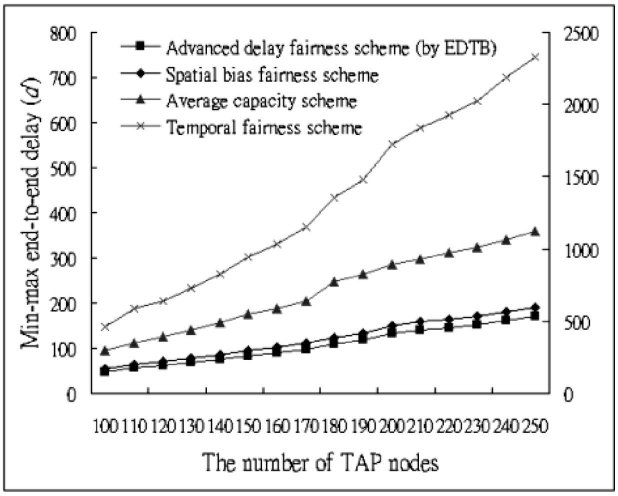


Fig. 9 Experiment results using different numbers of nodes.

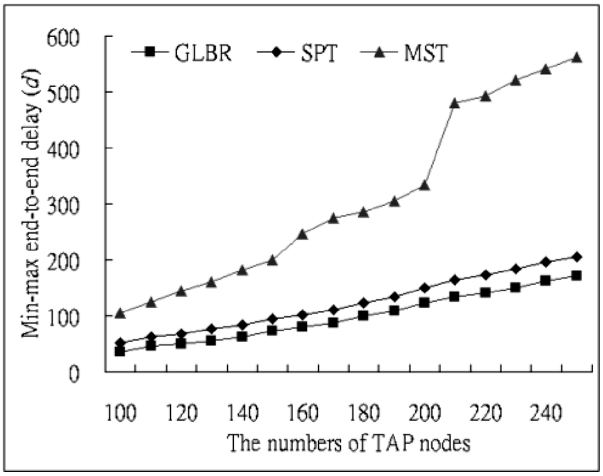


Fig. 11 Experiment results using different routing algorithms.

Figure 9 shows the experiment results for condition (ii) using different numbers of TAP nodes. We added the right-hand axis to indicate the the temporal fairness scheme only, as the delay time is much longer than that of the other schemes. The routing algorithm GLBR is used to achieve top-level load balancing. When the number of nodes increases, the maximal end-to-end delay is reduced because the traffic flow is distributed over a larger number of nodes. However, the end-to-end delay may increase as the traffic requirements and the number of hops increases. The proposed algorithm also outperforms the spatial bias fairness scheme by 10.19–13.26%. Even though the end-to-end delay achieved by the spatial bias fairness scheme is close to that of the extended delay fairness scheme (EDTB) and it achieves throughput fairness, the channel access time is unfair.

Figure 10 shows the experiment results for condition (ii) using different numbers of backhauls. As expected, the delay decreases as the number of backhauls increases. More backhauls provide more resources and reduce the network's traffic load. They also enable TAPs to route to closer backhauls with fewer hops and less delay (as described in experiment (i) in Fig. 8).

Figure 11 shows that the proposed routing algorithm outperforms general routing algorithms (e.g., the shortest path algorithm (SPT) and the minimum spanning tree (MST)) by 16.6–72.15%, especially when the number of nodes is larger. When the number of nodes is small, the number of hops and the traffic load is light. The difference in the performances of these algorithms is not significant, even though our proposed algorithm outperforms the other algorithms. However, when the number of nodes increases, the proposed GLBR algorithm tries to find the shortest path with load-balanced constraints. This improves the end-to-end delay significantly, even though the traffic requirement is same when the path length (the number of hops) is different.

6. Conclusion

We have compared different schemes for temporal, throughput, and bandwidth fairness. In a simple situation, end-to-end delay was found to be longer and the throughput for a node was lower on longer paths because of the larger amount of aggregated flow on those links. In addition, when advanced bandwidth allocation and min-max schemes are used with the proposed EDTB algorithm, end-to-end delay decreases and system throughput increases. First, we apply the proposed GLBR algorithm to obtain a load-balanced routing path. Then, we use the DFS traced nodes with the proposed EDTB bandwidth adjustment algorithm to calculate the allocated bandwidth and link transmission time for each node. Finally, the maximum end-to-end delay d is calculated. The experiment results show that the advanced delay fairness scheme not only achieves temporal and throughput fairness simultaneously, it also outperforms other schemes. (In terms of spatial bias or the average bandwidth per node it outperforms other approaches by at least 10.19% using different numbers of nodes).

Even though our proposed GLBR routing algorithm is a centralized approach, the routing of each node is based on its delay metric and the aggregated cost of the candidate subtree. The algorithm can be implemented with feedback messages or controlled by local candidate backhauls to select a lower backhaul delay and routing path. This issue could be refereed to Humblet's distributed algorithm for minimum-weight spanning trees [8]. The proposed EDTB bandwidth allocation algorithm can calculate the proportion of bandwidth for each TAP node in a shorter time based on the moving average traffic flow within a period. We will address these distributed algorithms in our future work.

References

[1] B. Li, "End-to-end fair bandwidth allocation in multi-hop wireless ad hoc networks," Proc. IEEE ICDCS, pp.471–480, Columbia, OH, June 2005.

[2] H. Dai and R. Han, "A node-centric load balanced algorithm for wireless sensor networks," Proc. IEEE GLOBECOM, pp.548–552, San Francisco, CA, USA, Dec. 2003.

[3] I.F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," Elsevier Computer Networks, vol.47, no.4, pp.445–487, March 2005.

[4] I. Maki, H. Shimonishi, T. Murase, and M. Murata, "Hierarchically aggregated fair queueing (HAFQ) for per-flow fair bandwidth allocation," IEICE Trans. Commun., vol.E89-B, no.2, pp.427–435, Feb. 2006.

[5] J. Jun and M. Sichitiu, "The nominal capacity of wireless mesh networks," IEEE Wireless Communications, vol.10, no.5, pp.8–14, Oct. 2003.

[6] J. Padhye, R. Draves, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," Proc. ACM MOBICOM, pp.114–128, Philadelphia, PA, Sept. 2004.

[7] N.B. Salem and J.P. Hubaux, "A fair scheduling for wireless mesh networks," Proc. IEEE WiMesh, LCA-CONF-2005-030, Santa Clara, CA, Sept. 2005.

[8] P.A. Humblet, "A distributed algorithm for minimum weight directed spanning trees," IEEE Trans. Commun., vol.31, no.6, pp.756–762, 1983.

[9] P.H. Hsiao, A. Hwang, H.T. Kung, and D. Vlah, "Load-balanced routing for wireless access networks," Proc. IEEE INFOCOM, pp.986–995, Anchorage, Alaska, April 2001.

[10] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks," IEEE Commun. Mag., vol.43, no.3, pp.123–131, March 2005.

[11] R. Karrer, A. Sabharwal, and E.W. Knightly, "Enabling large-scale wireless broadband: The case for TAPs," Computer Communication Review, vol.34, no.3, pp.27–32, 2004.

[12] R. Jain, W. Hawe, and D. Chiu, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report TR-301, Sept. 1984. http://www.cis.ohio-stateedu/˜jain/papers/fairness.htm

[13] T.J. Tsai and J.W. Chen, "IEEE 802.11 MAC protocol over wireless mesh networks: Problems and perspectives," Proc. IEEE AINA, pp.60–63, Taiwan, March 2005.

[14] V. Gambiroza, B. Sadeghi, and E. Knightly, "End-to-end performance and fairness in multihop wireless backhaul networks," Proc. ACM MOBICOM, pp.287–301, Philadelphia, PA, Sept. 2004.

[15] Y.F. Wen and Frank Y.S. Lin, "The top load balancing forest routing in mesh networks," Proc. IEEE CCNC, pp.468–472, Las Vegas, NV, Jan. 2006.

Yean-Fu Wen received the M.S. degree from the Department of Information Management, National Taiwan University of Science Technology, Taiwan, R.O.C., in 1998. In August 2000, he is pursuing the Ph.D. degree in the Department of Information Management, National Taiwan University. He also joined the Department of Information Management, China University of Technology. His research interests include network planning, performance optimization, and cross-layer technology in next-generation wireless networks. He is a student member of IEEE.

Yeong-Sung (Frank) Lin received his BS degree in electrical engineering from the Electrical Engineering Department, National Taiwan University in 1983, and his Ph.D. degree 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, Prof. 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.