Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 2450–2466
Computer Networks
www.elsevier.com/locate/comnet

Efficient load-balancing routing for wireless mesh networks

Yigal Bejerano [a], Seung-Jae Han [b,*,1], Amit Kumar [c]

[a] *Bell Laboratories, Lucent Technologies, NJ, United States*
[b] *Department of Computer Science, Yonsei University, Seoul, Republic of Korea*
[c] *Department of Computer Science and Engineering, Indian Institute of Technology Delhi, New Delhi, India*

Received 13 April 2006; accepted 25 September 2006
Available online 30 November 2006

Responsible Editor: E. Knightly

Abstract

Wireless mesh networks (WMNs) consist of static wireless routers, some of which, called *gateways*, are directly connected to the wired infrastructure. User stations are connected to the wired infrastructure via wireless routers. This paper presents a simple and effective management architecture for WMNs, termed *configurable access network* (CAN). Under this architecture, the control function is separated from the switching function, so that the former is performed by an *network operation center* (NOC) which is located in the wired infrastructure. The NOC monitors the network topology and user performance requirements, from which it computes a path between each wireless router and a gateway, and allocates fair bandwidth for carrying the associated traffic along the selected route. By performing such functions in the NOC, we offload the network management overhead from wireless routers, and enable the deployment of simple/low-cost wireless routers. Our goal is to maximize the network utilization by balancing the traffic load, while providing fair service and quality of service (QoS) guarantees to the users. Since, this problem is NP-hard, we devise approximation algorithms that provide guarantees on the quality of the approximated solutions against the optimal solutions. The simulations show that the results of our algorithms are very close to the optimal solutions.

Keywords: Wireless mesh network; Load-balancing routing; Fairness; Approximation algorithms

1. Introduction

The recent advance of wireless communication technologies has prompted a flourish of a new kind of multi-hop wireless network architecture, called *wireless mesh networks* (WMNs). WMNs typically comprise a number of static *wireless routers* that are attached to reliable sources of energy. The wireless routers are interconnected with each other via *wireless links* and provide communication services to mobile or static users in their vicinities. Some of the routers are directly connected to a fixed infrastructure (i.e., a wired network like the Internet) and serve as *gateways* for other wireless routers.

E-mail address: sjhan@cs.yonsei.ac.kr (S.-J. Han).
[1] This work is in part supported by KOSEF Grant R01200-60001086402006.

1389-1286/$ - see front matter © 2006 Published by Elsevier B.V.
doi:10.1016/j.comnet.2006.09.018

Commercial wireless mesh solutions are currently offered by such vendors as BelAir Networks (www.belairnetworks.com), Mesh Dynamics (www.meshdynamics.com) and Air Matrix (www.airmatrix.net), mainly for applications like wireless broadband access networks and disaster recovery networks. For effectively serving such applications, the network utilization must be maximized while providing fairness and bandwidth guarantees to the users [1].

One of the main concerns for WMNs is the reduction of the overall network capacity due to interferences between adjacent nodes [2]. To mitigate wireless interferences, several techniques can be used including multiple radios [2,3], directional antennas [4] and MIMO (multiple input multiple output). However, such physical layer solutions alone are not enough. To maximize the network utilization while preserving fairness requirements, efficient routing scheme is critical [1]. To address this need, we propose a simple and effective load-balanced routing scheme, via which the network utilization is maximized while providing fairness and bandwidth guarantees. The proposed scheme is compatible with any physical layer solution for interference mitigation.

1.1. Related work

Management of wireless multi-hop networks has been an active research area and numerous routing algorithms have been proposed. Comprehensive surveys on WMNs and routing in multi-hop wireless networks can be found in [5,6]. Most of the routing schemes for multi-hop wireless networks aim at such environments as battlefield ad-hoc networks, and the typical objective is to maintain the communication links between mobile stations. Providing connectivity, however, is not sufficient for WMNs in which the users demand Quality of Service (QoS) guarantees comparable to the wired networks.

Several routing schemes have been proposed for WMNs. Here, we mention only those studies that are directly relevant to our work. In [7], De couto et al. introduce the "expected transmission count" (ETX) metric that enables existing routing algorithms to find high performance paths between source–destination pairs where a single radio channel is used. Draves et al. propose in [2] a new path metric, called "weighted cumulative expected transmission time" (WCETT), that explicitly accounts for the interference among links using the same channel. Then, they incorporate the WCETT metric into a source-route link-state-like routing that exploits the advantage of the multiple radios. In [8,10] the authors address both the channel assignment and the routing problems of WMNs with multiple radios. They present channel assignment heuristics that maintain connectivity requirements while minimizing the interferences. In [8], Tang et al. present a bandwidth aware routing (BAR) algorithm that selects either a single path or multiple paths for each incoming session request, which maximizes bandwidth allocation of that session. In [10], Kyasanur and Vaidya present an enhanced shortest path routing method. Beside the hop-count of the path, it takes into account additional aspects such as the interferences from other nodes that use the same channel. In summary, the above-mentioned routing schemes increase the bandwidth allocation of individual session requests and typically achieve high overall network utilization. However, they do not consider the issue of fairness in the bandwidth allocation, so that wireless stations that are several hops away from the gateway may suffer from low bandwidth.

For the fair partition of the network resources, several studies use balanced trees rooted at the gateways and route the traffic along the tree paths. These studies commonly assume WMNs in which multiple radios are used while using different interference models. In [11], Hsiao et al. consider a grid mesh topology with a single gateway where each node has a interference-free point-to-point connection to its immediate neighbors. Under this setting, a centralized heuristic is proposed for calculating load-balanced H-trees that allocate the same bandwidth to all the nodes. In [12,3], both channel assignment and routing are considered together. In [12], He et al. present a heuristic for calculating load-balanced shortest path tree by taking the traffic flows into account. After calculating the tree, they perform channel assignment to efficiently utilize the wireless links. In [3], Raniwala and Chiueh present distributed algorithms that use only local traffic load information for determining the channel assignment and the tree topology. Though these existing heuristics improve the fairness of the allocated bandwidth to the nodes, they do not provide any guarantee on the quality of the solutions against the optimality.

Some studies exploit the advantage of multiple-path routing. In [13], Jain et al. consider the problem of optimal multi-path routing, where the interferences are modeled by a conflict graph. A similar problem is addressed by Kodialam and Nandagopal in [14]. This study deals with the joint problem of routing and scheduling of multi-path flows, assuming that each wireless station is equipped with a single radio but the stations use orthogonal channels in order to avoid interferences. In [15], the authors extend their result for the case of multiple radios. These studies have shown that multi-path routing maximizes overall traffic flow while providing fair service and bandwidth guarantees. However, these methods face difficulties in the traffic management, since the traffic between each source–destination pair may be divided into multiple small flows and they generate high communication and computation overhead on the network nodes [16,17]. In [17], Ganjali and Keshavarzian claim that in practice the load distribution obtained by multi-path routing is essentially similar to the single path routing, unless a very large number of paths are used (which is practically infeasible).

1.2. Our contributions

In this paper, we present a simple and effective management architecture for WMNs, termed *configurable access network* (CAN). This architecture is inspired by the observation that WMNs usually serve as access networks [1,3] and consequently all nodes of WMNs are accessible from the wired infrastructure. Such WMNs can be managed by external stations, termed a *network operation center* (NOC). The use of external NOCs offloads the management overhead from the wireless routers and thus reduces their complexity. This centralized management approach that separates the switching functions from the control functions have already been adopted in wired networks, e.g., the softswitch model of converged networks [18] and the softrouter architecture for IP networks [19]. In practice, some rudimentary centralized management schemes with external NOCs have already been adopted by some WMN commercial products such as BelAir Networks and Air Matrix.

We present algorithms for single-path routing and bandwidth allocation that can be incorporated into the CAN architecture. Our algorithms achieve near-optimal fair bandwidth allocation without the drawbacks of the multi-path routing. The single-path approach has several advantages over the multi-path approach: (i) it simplifies the traffic control, (ii) it maintains the packet delivery order, (iii) it enables the deployment of efficient compression schemes. Compression schemes, such as RObust Header Compression (ROHC) scheme [20], yield significant saving in bandwidth consumption by increasing network throughput without requiring the actual increase of the network capacity. However, high compression ratio can be obtained only when the packets maintain their order and traverse through the same gateway, which is not always possible in case of multi-path flows. Via extensive simulations, we examine the performance of the singe-path approach against the multi-path optimal solutions, without considering the gain by compression. The simulations show that the bandwidth allocation obtained by our single-path approach is very close to the multi-path optimal solution, in particular when the number of nodes and users is high. This confirms the observation of Ganjali and Keshavarzian in [17]. Our simulations show that the selected paths are typically along the shortest paths between the nodes and the gateways, resulting minimal end-to-end delay which is critical for real-time applications.

Our algorithms find a set of paths that maximize the *normalized bandwidth allocation* of the nodes, which is defined as the minimal bandwidth allocated to each user. This objective is aligned with the fairness reference model described in [21], that aims at the fair bandwidth allocation to the users independent of their locations or their distances from a gateway. Since this problem is NP-hard, we develop several algorithms that provide different approximation ratios under different settings. All these algorithms initially calculate the optimal multi-path flow solutions. Then, we utilize the single-source unsplittable flow algorithm by Dinitz et al. [22] to extract the single-path routing from the multi-path solutions. Each wireless router (termed a node) is associated with a weight d_v, that represents the number of users who are associated with the node. We first present an algorithm that finds an optimal solution when all nodes have the same weights and all wireless links have the same capacity. When the nodes have different weights, this algorithm guarantees at least half of the optimal normalized bandwidth. Then, we deal with the case that links can have arbitrary capacities and the node weights are bounded by $d_{\min}$ and $d_{\max}$. In this case, we achieve a $1+\alpha$-approximation factor, where $\alpha = \frac{d_{\max}}{d_{\min}}$, by forwarding the traffic only along links with sufficient capacity. This is actually 2-approximation for the fair aggregated bandwidth allocation objective presented in [21]. For the case of arbitrary node weights and link capacities, we provide a 5-approximation algorithm. To the best of our knowledge, our scheme is the first work that provides guarantees on the quality of single-path routing solutions in the context of wireless multi-hop routing.

Our goal is maximizing the network utilization, while providing fairness. Since, wireless channel quality may frequently change due to interferences and fast fading effects [9], it is a daunting task to achieve this goal in a short-term scale, i.e., to optimize the network performance at any given moment. Frequent changes of the traffic routes may lead to route oscillations which can severely degrade the overall performance. We focus on the *long-term fairness* by occasionally modifying the traffic routes to address topology changes or user mobility.

2. Network model

This paper considers WMNs which comprise of static or quasi-static wireless mesh routers, termed *nodes*, and mobile or static user stations, termed *users*. Some nodes, referred as *gateways*, are equipped with *backhaul links* to a fixed infrastructure network and serve as access gateways for other nodes. Each node is typically equipped with one or more omni-directional antennas (e.g., using IEEE 802.11-b or g) to provide the network connectivity to the users in its vicinity. It also has additional backhaul radio interfaces for point-to-point connectivity with its adjacent nodes (e.g., using IEEE 802.11-a or IEEE 802.16). The backhaul radio interfaces may use directional antennas.

The network is represented by an undirected graph $G(V \cup \{a\}, E)$, where the graph nodes, V, represent the wireless routers and a is a virtual node that corresponds to the fixed infrastructure. We denote by E the network links including the backhaul links between the gateways and the infrastructure as well as the wireless links between nodes. Each link $e \in E$ has a capacity (bit-rate) C_e for both direction. Since, a channel quality may be time-varying, we assume the C_e is the average link capacity or its lower bound, depending on the QoS requirements. For instance, for delay-tolerant applications C_e may use the average channel quality, while for real-time applications the lower bound may be used. By assuming directional antennas and sufficient number of wireless channels,[2] we ignore the interference between channels.

Each node $v \in V$ is associated with a weight $d_v \in \mathscr{Z}^+$ that is proportional to its bandwidth requirement. Typically, d_v indicates the number of users associated with node v. Let b_v be the *bandwidth allocation* for node v. b_v/d_v is the average bandwidth allocated to each one of its associated users and referred as the *normalized bandwidth*, $\overline{B}$ to the minimal bandwidth that a user may experience, i.e., $\overline{B} = \min_{v \in V} b_v/d_v$. We denote by $d_{\max} = \max_{v \in U} d_v$ and $d_{\min} = \min_{v \in U} d_v$ the maximal and minimal weights of the nodes, respectively. The *neighborhood* of node $v \in V$, denoted by $N(v)$, is the set of nodes with whom v has connections. When node v is a gateway. the virtual node a is included in the neighborhood of v. In this paper, we assume that the user traffic always flows from or to the fixed infrastructure, so that the traffic always traverses through the gateways, while non-gateway nodes serve as relays.

3. The CAN architecture

One of the main challenges in WMNs is achieving *high network utilization* and *long-term fairness* simultaneously. This objective is crucial as nodes that are several hops away from their serving gateways may experience low service quality or even been starved. It requires efficient routing and bandwidth allocation. The algorithms for route selection and bandwidth allocation will be treated in Section 4, and this section describes the CAN architecture which can facilitate such algorithms.

3.1. Overview

The CAN architecture is inspired by the following observation: WMNs are mainly used as access networks for sending/receiving information to/from the users via wireless routers. As all nodes are always connected to a wired infrastructure, it is possible to shift the resource management tasks, that may be done by the nodes, to a NOC, which is located in the wired infrastructure. In particular, we propose that the NOC determine the routes between the nodes and the gateway as well as allocating appropriate bandwidth for each traffic flow.

[2] For instance, eight (or more) non-overlapping channels are available for IEEE 802.11-a.

To this end, the NOC needs to know the network topology, the link capacities and the nodes' weights. Based on the collected information, it determines routing and bandwidth allocation, and configures the nodes accordingly.

In WMNs the wireless nodes are typically static (or quasi-static) and connected to reliable sources of energy. Such networks experience relatively infrequently topology changes that can easily be reported to the NOC. The CAN architecture has several advantages over the existing methods for WMNs that are typically based on distributed self-organized solutions. In contrast to the existing solutions that mostly focus on maintaining connectivity, our solution maximizes the fair bandwidth allocation to the users. Moreover, since the routing and bandwidth allocation decisions are made at the NOC, the wireless routers need only modest computation capabilities. The CAN architecture also reduces the communication overhead, since topology changes are informed only to the NOC instead of broadcast to all other nodes. These benefits make the CAN architecture particularly attractive for large-scale WMNs.

3.2. Inferring the network topology

Consider a wireless mesh network modeled by a graph $G(V \cup \{a\}, E)$ as described in Section 2. All nodes are assumed to know their immediate neighbors and the NOC is connected to all the gateways. Upon the activation, a NOC starts to infer the network topology, the link capacities, and the node weights by sending out queries. At first, by querying the gateways, the NOC obtains the addresses of all the nodes that are one hop away from the gateways. Similarly, by querying all the gateways' neighbors, the NOC learns about all the nodes that are two hops away from the gateways. Thus, by performing breadth first search (BFS) the NOC discovers the network topology layer by layer. During this topology discovery process, the NOC communicates with a node by using source-routing method, i.e., each control message carries its complete path to its destination and the reply is returned on the reversed path.

3.3. Single vs. multiple traffic paths

One of the main decisions that the NOC makes is computing the traffic routes for each individual user. Either a single path or multiple paths can be used for each user traffic flow. Generally speaking, the multiple-path approach maximizes the network utilization and the fair share of each user. This approach is also more resilience to failures. The price for such benefits is the complexity in the traffic management. For instance, it may require dividing the packets of a single flow to multiple routes, managing large number of small flows and maintaining the packets order at the aggregation point. Therefore, we believe that routing the traffic of each flow along a single path is a more pragmatic approach. In a single-path approach, for each flow, the NOC selects a single path, termed a *virtual connection* (VC), that has a dedicated bandwidth allocation and a unique VC identifier.[3] Packets of this flow carries in their header the VC identifier, which is used by the intermediate nodes to forward the packets along the selected path. Such packet forwarding method is used, for instance, by ATM [23] and MPLS [24].

In addition to the simplicity in the network management and traffic control, the single-path approach has other important advantages over multiple-path approach. It maintains the packet delivery order, which is important to the performance of many protocols such as TCP. Maintaining packet order is in particularly crucial for the use of header comparison algorithms. For instance, in the case of voice over IP (VoIP) traffic, maintaining the packet order is essential for efficient utilization of the Robust Header Compression (RoHC) protocol [20], which reduces the packet size from 62 bytes (82 bytes in the case of IPv6) to only 24 bytes, assuming voice payload of 22 bytes per packet. Such compression can significantly increase the network utilization. Furthermore, our simulation results in Section 5 show that in practical settings, efficient single-path solutions yield network utilization and normalized bandwidth similar to the optimal solutions using multiple path flows. Taking the gain by header compression into account, the single-path solutions can outperform the multiple-path solutions.

In this paper we consider two variants of single-path solutions. In the first variant, termed a *aggregated flow* model, the aggregated traffic of all the users that are associated with a given node v is forwarded along a single path. While, the second variant, termed a *user flow* model, the traffic of each

[3] By using label swapping technique, each VC need to have unique VC identifier at each node.

user is routed separately over its own path. On one hand, the user flow model may utilize the network resources more efficiently thanks to its multiple-path nature. It may be beneficial in particular for applications of constant bit-rate user flows like voice calls. On the other hand, since each user has a dedicated bandwidth allocation, the user flow model may be less desirable for the applications that have bursty traffic characteristics. For such bursty traffic, the aggregated flow model may be more efficient by utilizing statistical multiplexing among multiple users that share the same VC.

The single-path solution is more vulnerable to the network failures. This problem can be overcome by setting up one or more backup VC(s) for each flow, and in case of a failure the affected flows are rerouted along the backup baths. Since the nodes are connected to reliable source of energy, we expect failures rarely occur. In this paper we focus on the routing of the primary VCs. An example of an aggregated flow route selection is given in Example 1.

Example 1. An example of a CAN system with two gateways b and c, is depicted in Fig. 1. In this example the gateways are attached to a virtual node a that represents the fixed infrastructure. We assume that all the links have capacity of 6 and the weights (i.e., bandwidth demands) of the nodes are $d_b = 3$, $d_g = 4$, $d_h = 3$ and $d_i = 2$. For simplicity, we assume that the weights of the other nodes $\{c, d, e, f\}$ are zero. The figure shows the optimal route selection where both backhaul links (a, b) and (a, c) are full utilized and the normalized bandwidth is $\overline{B} = 1$.

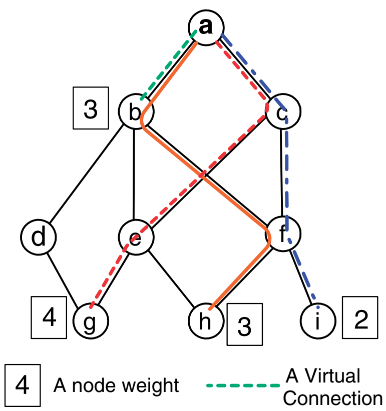

Fig. 1. An example of a CAN system that utilizes aggregated flow route selection.

3.4. Network configuration

After computing the set of VCs and determining the bandwidth allocation of each VC, the NOC configures the nodes accordingly as follows: first, it builds a forwarding table for each node $v \in V$. This table contains a record for every VC that traverses through node v with the required information for packet forwarding, such as the VC label, the identifications of the predecessor and successor nodes and the VC bandwidth allocation. Second, it updates the node forwarding tables by sending dedicated configuration messages. Finally, it sends activation messages to the nodes to start data forwarding.

Due to the user mobility, the node weights and their bandwidth demands may change. Such changes may require modifications of the VC bandwidth allocations. Since nodes typically support multiple users, the movement of a user typically incurs only small weight changes that can be easily addressed by minor adjustments of the VC bandwidth allocations, without changing the VC routes. When the network configuration is significantly deviated from the optimal settings, the NOC recalculates the nodes' VCs and reconfigures the network. To maintain the continuous network operation during the configuration setting operation, dual sets of VCs are used. At each configuration setting operation the NOC modifies one VC set, while the other set is used to forward traffic. At the end of the operation, the NOC send a message to all nodes to switch to the alternative set. We do not address the online bandwidth allocation change any further, and focus on the optimal route selection.

4. Single-path routing algorithms

Our single-path routing algorithms (for primary paths) maximize the normalized bandwidth of the nodes. The problem formulation is given in Section 4.1. We consider two route selection settings. The first setting, referred as the *aggregated flow* model, aggregates traffic of all the users that are associated with a given node v and routes the aggregated flow along a single path. For this setting, we present three algorithms that provide different guarantees on the quality of the solution depending on the characteristics of the underlying networks. In Section 4.2, we present a polynomial time algorithm that finds an optimal route selection when all nodes have the same weight and all links have the same capacity. This algorithm also ensures a solution with at least half the normalized bandwidth of the optimal integral solution, when all the links have the same capacity but nodes may have arbitrary weights. In Section 4.3 we introduce the concept of *relay group* and we use it to construct an $(1+\alpha)$-approximation for the case of bounded node weights and arbitrary link capacities, where α is the ratio between the upper and lower bounds on the node weights. Then, in Section 4.4, we present a 5-approximation algorithm for the general case. In the second setting, termed an *user flow* model, users who are associated with the same node are not required to share the same path, and the traffic of each individual user is routed separately (still a single path for each user). For this model, we present an algorithm that ensures 2-approximation in Section 4.5.

4.1. The problem statement

Consider a graph $G(V \cup \{a\}, E)$ as described in Section 2. We denote by V the set of nodes (i.e., wireless routers), while a is a virtual node that represents the fixed infrastructure. Let E be the set of edges comprising both backhaul links and wireless links between adjacent nodes. Each link $e \in E$ is associated with capacity C_e and d_v is the weight of a node $v \in V$. We tackle the problem of *route selection and bandwidth allocation* that maximizes the minimal bandwidth allocated to each user, termed the *normalized bandwidth*. More specifically, let P_v be the selected route for a flow originated by node v (or by a user associated with node v) and let b_v be the allocated bandwidth to this flow. Then the *normalized bandwidth* $\overline{B}$ is defined as the guaranteed minimal bandwidth to each user, i.e., $\overline{B} = \min_{v \in V \wedge d_v > 0} b_v/d_v$. A route selection is termed *feasible* if the overall bandwidth allocation of all the routes that traverse through any given link $e \in E$ does not exceed the link capacity C_e. We formally define the route selection problem as follows:

Definition 1 (*The bandwidth maximization problem*). Given a graph $G(V \cup \{a\}, E)$ with capacity C_e for every link $e \in E$ and weight d_v for every node $v \in V$. Find a feasible set of paths $\{P_v\}$ and the corresponding bandwidth allocations $\{b_v\}$ between a and every node $v \in V$, that maximize the normalized bandwidth, i.e., $\overline{B} = \max \min_{v \in V \wedge d_v > 0} b_v/d_v$. The selected set of paths $\{P_v\}$ and the corresponding bandwidth allocations are termed a *route selection*.

Theorem 1. *The bandwidth maximization route selection problem is NP-hard in the case of different node weights.*

Proof. We prove this theorem by presenting a polynomial reduction from the *partition* problem [25] to the route selection problem. Consider a set $\mathcal{Q}$ of $m > 2$ elements with size $s_i \in \mathcal{Z}^+$ for every $q_i \in \mathcal{Q}$, and let $X = \sum_{q_i \in \mathcal{Q}} s_i/2$. The partition problem looks for a sub set $\mathcal{Q}' \subset \mathcal{Q}$ such that $\sum_{q_i \in \mathcal{Q}'} s_i = \sum_{q_i \in \mathcal{Q}-\mathcal{Q}'} s_i = X$. We construct a graph $G(V \cup \{a\}, E)$ with a node v_i for every element $q_i \in \mathcal{Q}$, and two gateways denoted by u_1 and u_2, as depicted in Fig. 2. The weights of the nodes u_1 and u_2 are 0, and the weight of every node v_i is q_i, i.e., $d_{v_i} = q_i$. Moreover, let assume that the capacity of the backhaul channels (u_1, a) and (u_2, a) are $C_{u_1,a} = C_{u_2,a} = 2 \cdot X$. We claim that there is a subset $\mathcal{Q}' \subset \mathcal{Q}$ with $\sum_{q_i \in \mathcal{Q}'} s_i = X$ if and only if there is a feasible route selection with normalized bandwidth of 1. Suppose that there is such a partition $\mathcal{Q}'$. Then we construct the routes as follows, for every $q_i \in \mathcal{Q}'$ the path P_{v_i} between the virtual node a and node v_i traverses through node u_1, otherwise the path traverse through node u_2. Thus, the aggregated weights of all the nodes that are served by each one of the gateways u_1 and u_2 is $X/2$. Thus, the normalized bandwidth allocated to all nodes is 1. Now, suppose that there is a feasible route selection, such that the normalized bandwidth is 1. Let $\mathcal{Q}'$ be the set of elements represented by all the nodes that their routes traverse trough node u_1. So the sum $\sum_{q_i \in \mathcal{Q}'} s_i = X$ and this completes the proof. □

4.2. The basic scheme

We now turn to describe our route selection algorithms for the aggregated flow model. All the proposed algorithms consist of two steps. In the first step, the algorithms calculate a *fractional solution*, also termed a *splittable flow solution*, that is allowed to divide each traffic flow b_v into small flows and route them along multiple paths. The fractional routing problem is formulated as a single-source

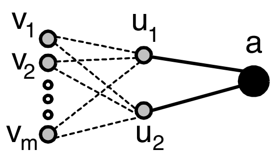

Fig. 2. The graph for the proof of Theorem 1.

multiple-destinations flow problem that maximizes the normalized bandwidth of the users. Recall that such fractional solution always exists. Then, we utilize a *rounding algorithm* for obtaining an *integral solution*, termed an *unsplittable flow solution*, with a single route for each node flow. The latter is based on the single-source unsplittable flow algorithm of Dinitz et al. [22]. Since the rounding algorithm may calculate a link total flow that exceed the link capacity, after the rounding step we scale the bandwidth allocations to satisfy the capacity constraints. Note that the fractional solution is an upper bound for the optimal integral solution.

4.2.1. The fractional solution

We start with a scheme that allows all the nodes to serve as relays. Consider a graph $G(V \cup \{a\}, E)$ as described above and let b_v be the bandwidth allocation of every node $v \in V$. We denote by $F_{u,v}$ the flow along the link (u,v) from node u to node v and let $C_v^{\max} = \max_{u \in N(v)} C_{u,v}$ be the maximal capacity of any incoming link of node v. Our goal is maximizing the normalized bandwidth allocation defined by $\overline{B} = \min_{v \in V \wedge d_v > 0} b_v / d_v$. Thus, the fractional routing problem can be formulated as a linear program (LP) as follows:

$$\max \ \overline{B}$$

subject to

$$\forall v \in V \wedge d_v > 0 : \overline{B} \leqslant b_v / d_v, \qquad (1)$$

$$\forall v \in V \wedge d_v = 0 : b_v = 0, \qquad (2)$$

$$\forall v \in V : \sum_{u \in N(v)} F_{u,v} = \sum_{u \in N(v)} F_{v,u} + b_v, \qquad (3)$$

$$\forall (u,v) \in E : F_{u,v} + F_{v,u} \leqslant C_{u,v}, \qquad (4)$$

$$\forall v \in V : b_v \leqslant C_v^{\max}, \qquad (5)$$

$$\forall (u,v) \in E : F_{u,v} \geqslant 0, \quad F_{v,u} \geqslant 0. \qquad (6)$$

In this formulation, Constraints (1) and (2) ensure that $\overline{B}$ lower bounds the normalized bandwidth allocation of every node $v \in V - \{a\}$ with positive weight, while bandwidth is not allocated to nodes with weight zero. Constraint (3) is the *flow conservation requirement* that ensures that the amount of flow withdrawn by node v is exactly its bandwidth allocation b_v. Constraint (3) also ensures that the aggregated flow originated by the source node a is $\sum_{v \in V - \{a\}} b_v$. Constraint (4) guarantees *capacity constraint*. Finally, constraint (5) is an *allocation constraint* to guarantee an upper bound on a node allocated bandwidth, such that a node bandwidth allocation does not exceed the maximal capacity of its edges. The optimal fractional solution can be found by using any LP solver or a maximal flow approach, as described in [26]. Alternatively, approximation methods, like the ones described in [27] can be used to find near-optimal solutions. It is easy to see that

Corollary 1. *The solution of the fractional routing problem is an upper bound of the normalized bandwidth allocation and it can be calculated in a polynomial time.*

Example 2. Consider the CAN system with two gateways, b and c, as described in Example 1. In this system all the links have the same capacity 6 and the node weights are $d_b = 3$, $d_g = 4$, $d_h = 3$ and $d_i = 2$. For simplicity, we assume that the weights of the other nodes $\{c,d,e,f\}$ are zero. Fig. 3(a) presents the fractional flow solution for the given graph. Recall that the flows through the links (a,b) and (a,c) are $F_{a,b} = F_{a,c} = 6$. Thus, the normalized bandwidth allocation to all the nodes is $\overline{B} = 1$. In other words, $b_b = 3$, $b_g = 4$, $b_h = 3$ and $b_i = 2$.

4.2.2. The rounding algorithm

After calculating a fractional routing, we round the fractional flows to obtain an integral solution. Let $G'(V,E')$ be the directed graph induced by the flows of the fractional solution. A directed link $(u,v) \in E$ is included in G' only if there is strictly positive flow from u to v. Without loss of generality, we assume that $G'(V,E')$ is acyclic graph, as directed cycle can be eliminated by flow decomposition. Now, we utilize the single-source unsplittable flow algorithm of Dinitz et al. on the constructed graph

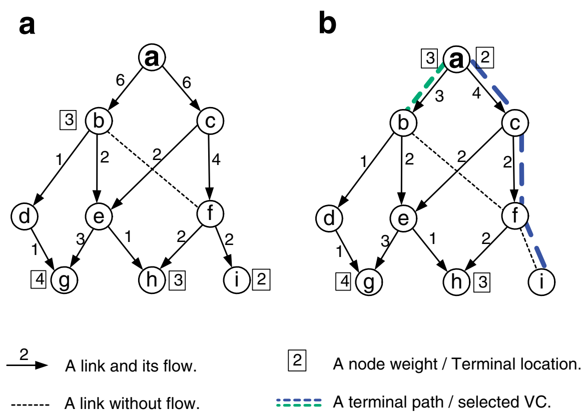

Fig. 3. The fractional solution (a) and the preliminary terminal shifting phase (b).

G'. For completeness, we provide a concise description of the algorithm. Its correctness and performance analysis can be found in [22].

The rounding algorithm associates a token t_v with each node $v \in V - \{a\}$ that represents a traffic flow of size b_v. The algorithm modifies the network flows by moving the tokens backward, until they reach the source node a. As token t_v is moved backward along an edge e, the flow F_e is reduced by b_v and edges with zero flow are eliminated. So, at any time the network satisfy the flow conservation requirement in respect to the current locations of the tokens. Finally, the algorithm selects the movement route of the token t_v as the route P_v of every node v.

Let us denote by t_v both the token identifier and its current location. In a preliminary phase, the algorithm checks for every token t_v whether there is an incoming edge $e = (u, t_v)$ with flow greater than or equal to node's v allocated bandwidth b_v. In such case, it move t_v to u and decreases the flow of e by b_v. If e does not carry any more flow it is removed from the graph. The algorithm repeats this process as much as possible and retains only tokens that do not coincide with the source node a. Observe that the resulting instance maintains a *degree property* such that the tokens are located only at nodes with at least two incoming edges. An example of the preliminary phase is given in Fig. 3(b) for the fractional solution presented in Fig. 3(a).

The rounding algorithm proceeds in iterations that each one consists of three steps. It first finds an *alternating cycle*, then it augments the flow along this cycle and finally it shifts tokens according to a movement rule that keeps the degree property. The algorithm constructs an alternating cycle by performing a tour on the graph edges. It starts at the source node a and creates a *forward path* by following outgoing edges as long as possible. Since the graph is acyclic, the forward path must end at a node with a token t_v. The algorithm proceeds by constructing a *backward path* starting from t_v. Since t_v has at least two incoming edges, the algorithm chooses an unselected edge[4] and follows the incoming edge until reaching the first node, say u, that has another outgoing edge. The algorithm builds another forward path by following this outgoing edge of u. The process continues in this manner until it reaches a node, say w, that has already been visited and close a cycle. Thus, the cycle consists of alternating forward and backward paths, as depicted in Fig. 4(a) and (c).

Now, the algorithm modifies the flow along this cycle by shifting a small amount of flow from the forward paths to the backward paths in a way that maintains the flow conservation requirement. It calculates two quantities, ϵ_f and ϵ_b. Where, ϵ_f is the minimal flow of the edges along the forward paths and ϵ_b is the minimal difference between the flow along a link (u, t_v) and the bandwidth allocation b_v for every token t_v that is located in one of the cycle nodes (or infinity if there are no tokens in the cycle). Then, the shifted amount of flow is the smaller of the two, i.e., $\min(\epsilon_f,\epsilon_b)$. Recall that if the minimum is achieved for ϵ_f then after the augmentation there is no flow along one of the forward edge and this

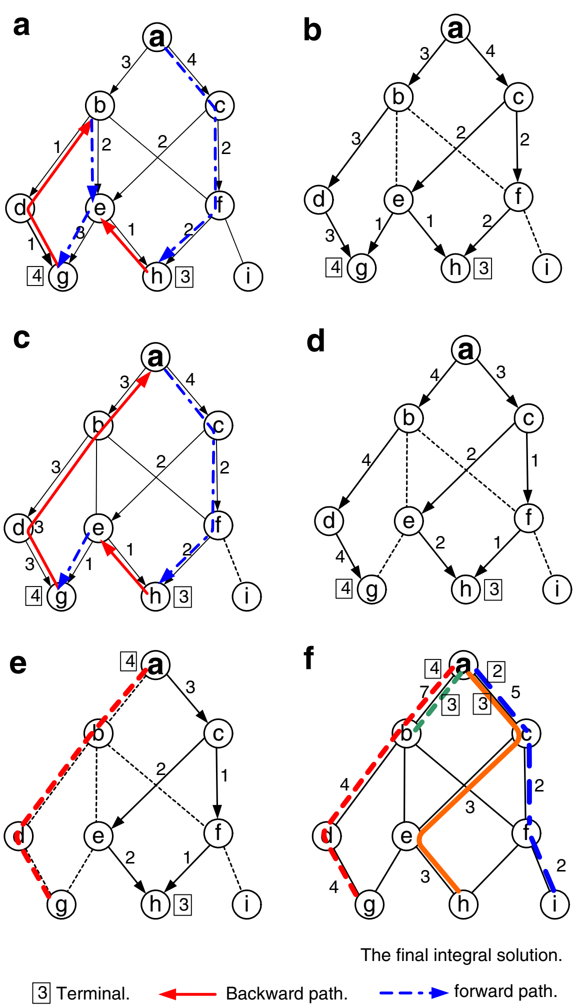

Fig. 4. An example of calculating an integral solution.

[4] An edge that is not included in the forward paths.

edge is removed. Otherwise, the minimum is obtained for an edge (u, t_v) on a backward path. After the augmentation, the flow along this path is b_v and the algorithm repeats the preliminary phase. So, the token t_v is moved along the edges of the backward path (and possibly more edges) and these edges are removed from the graph. The algorithm ends when all the tokens reach the AP and the corresponding paths are determined for the traffic flows. An execution of the algorithm is illustrated in Example 3.

Example 3. Fig. 4 illustrates few of the steps made by the rounding algorithm, while calculating an integral solution for the fractional solution given in Fig. 3(a). Fig. 4(a) and (c) presents two calculated alternating-cycles. While, Fig. 4(b) and (d) shows the resulting flow after shifting some flow from the forward paths to the backward paths (2 units in Fig. 4(b) and 1 unit in Fig. 4(d)). After performing those two flow shifting operations the token t_g can be moved to the virtual node a, as shown in Fig. 4(e), and the final integral solution is given in Fig. 4(f).

In this solution, link (a, b) serves the flows of both nodes b and g. Thus the aggregated weight of the flows that traverse through the link (a, b) is 7. However, as we calculated in Example 2, the normalized bandwidth of the optimal fractional solution is $\overline{B} = 1$. Thus, if we retain a normalized bandwidth of 1, the integral solution exceeds the capacity of the link (a, b). We overcome this over subscription problem by scaling down the normalized bandwidth, as we describe below. In this case, the selected normalized bandwidth is $\overline{B} = \frac{6}{7}$ and the bandwidth allocations are $b_b = 2\frac{4}{7}$, $b_g = 3\frac{3}{7}$, $b_h = 2\frac{4}{7}$ and $b_b = 1\frac{5}{7}$. As a result, the aggregated flow that traverses through the link (a, b) is 6 and it satisfies the link capacity.

Example 3 demonstrates that the rounding algorithm may not find the optimal integral solution. In Theorem 2 we bound the deviation from the optimal solution.

Theorem 2 (From [22]). *The rounding algorithm finds unsplittable flow such that the total flow through any edge exceeds its initial flow by less than the maximal allocated bandwidth,* $b_{\max} = d_{\max} \cdot \overline{B}$.

From Theorem 2 follows that after selecting a route P_v for each node $v \in V$, some of the link flows may exceed the link capacities. We overcome this *over subscription* problem by scaling down all bandwidth allocations. Let λ be the maximal *over subscription ratio*, i.e., $\lambda = \max_{e \in E} F_e / C_e$. For satisfying the capacity constraints we scale down all the bandwidth allocation by a factor of λ, such that the new bandwidth allocation of every node $v \in V$ is $b_v^{\mathrm{i}} = b_v / \lambda$.

4.2.3. Algorithm analysis

We analyze the quality of the calculated integral solution relative to an optimal one for a given graph $G(V \cup \{a\}, E)$. In our analysis, we consider the normalized bandwidth $\overline{B}^{\mathrm{f}}$, $\overline{B}^{\mathrm{i}}$ and $\overline{B}^{*}$ of the fractional, integral and optimal solutions, respectively. We measure the link flows of a given solution with units of its normalized bandwidth allocation $\overline{B}$, termed *normalized bandwidth unit*, i.e., $\overline{B} = 1$. Thus, the flow F_e of a link $e \in E$ represents the number of normalized bandwidth unit that traverse through this link. Obviously, the normalized bandwidth of a solution is maximized when the maximal link flow is minimized. Consequently, we denote by F_e^{f}, F_e^{i} and F_e^{*} the flows that traverse through link e in the fractional, integral and optimal solutions measured with the corresponding normalized bandwidth units, $\overline{B}^{\mathrm{f}}$, $\overline{B}^{\mathrm{i}}$ and $\overline{B}^{*}$, respectively. Hereby, we consider instances where all links have the same capacity.

Theorem 3. *If the weight of every node $v \in V$ is either zero or a constant d and all the links have the same capacity, C, then the basic algorithm finds an optimal integral solution.*

Proof. Let us consider the optimal fractional solution and let link $e' \in E$ be the one with the maximal flow, i.e., $F_{e'}^{\mathrm{f}} = \max_{e \in E} F_e^{\mathrm{f}}$. We now prove that both in the optimal and the integral solutions the most congested links carry flows of $\lceil F_{e'}^{\mathrm{f}} \rceil$ normalized bandwidth units. From Theorem 2 results that after the rounding algorithm the maximal flow of every link e is *less* than $F_e^{\mathrm{f}} + \overline{B} = F_e^{\mathrm{f}} + 1$. Since, each flow contains an integer number of normalized bandwidth units results that the maximal flow in every link is at most $\lceil F_{e'}^{\mathrm{f}} \rceil$. Now recall that e' is a bottleneck link in the fractional solution. Thus it is a lower bound on the number of normalized bandwidth units that are supported by the most congested link in the optimal solution. Since in the optimal solution each link supports integral number of normalized bandwidth units, follows that its most congested link carries a flow of at least $\lceil F_{e'}^{\mathrm{f}} \rceil$ normalized bandwidth units. This proves that both the calculated integral and optimal solutions have the same normalized bandwidth unit, which complete our proof. □

Theorem 4. *If all the links have the same capacity C then the basic algorithm guarantees a 2-approximation ratio. In other words,* $\overline{B}^{\mathrm{i}} \geqslant \overline{B}^*/2$.

Proof. We denote by $d_{\max}$ the maximal weight of a node and let e and l be the links with the maximal flows in the integral and the optimal solutions. Clearly both links have flows of at least $d_{\max}$ units of the corresponding normalized bandwidth units. From Theorem 2 follows that $F_e^{\mathrm{i}} < F_e^{\mathrm{f}} + d_{\max} \leqslant F_l^* + d_{\max}$. Since, $F_l^* \geqslant d_{\max}$ results that $F_e^{\mathrm{i}} < 2 \cdot F_l^*$. Thus,

$$\overline{B}^{\mathrm{i}} = \frac{C}{F_e^{\mathrm{i}}} \geqslant \frac{C}{2 \cdot F_l^*} = \frac{\overline{B}^*}{2},$$

which completes the proof. □

4.3. The bounded weight scheme

We now present the *bounded weight scheme* for networks where the weight d_v of very node $v \in V$ is bounded between $d_{\min}$ and $d_{\max}$ and we allow arbitrary link capacities, C_e. For such instances, our scheme ensures that the calculated normalized bandwidth, $\overline{B}^{\mathrm{i}}$, is at least $1/(1+\alpha)$ of the optimal solution, where $\alpha = d_{\max}/d_{\min}$. Thus, when all the nodes have the same demand we get 2-approximation ratio. This scheme also calculates first a fractional solution and then uses the rounding method described in Section 4.2.2. Unlike the basic scheme, this algorithm uses a different fractional routing formulation, in which only links with enough capacity carry traffic. We refer to this set of links as the *relay group*. Consider any integral solution and let $\overline{B}$ be it normalized bandwidth allocation. In this solution the traffic flows may only traverse over links with capacity $C_e \geqslant d_{\min}\overline{B}$. Thus, for a given normalized bandwidth $\overline{B}$ we define the network relay group to be the set,

$$R(\overline{B}) = \{e | e \in E \wedge C_e \geqslant \overline{B} \cdot d_{\min}\}. \tag{7}$$

We formulate the fractional routing problem as follows:

$$\max \overline{B}$$

subject to

$$\forall v \in V \wedge d_v > 0 : \overline{B} \leqslant b_v/d_v, \tag{8}$$
$$\forall v \in V \wedge d_v = 0 : b_v = 0, \tag{9}$$
$$\forall v \in V : \sum_{u \in N(v) \wedge (u,v) \in R(\overline{B})} F_{u,v} = b_v + \sum_{u \in N(v) \wedge (u,v) \in R(\overline{B})} F_{v,u}, \tag{10}$$
$$\forall (u,v) \in R(\overline{B}) : F_{u,v} + F_{v,u} \leqslant C_{u,v}, \tag{11}$$
$$\forall (u,v) \in R(\overline{B}) : F_{u,v} \geqslant 0, \quad F_{v,u} \geqslant 0, \tag{12}$$
$$\forall (u,v) \notin R(\overline{B}) : F_{u,v} = 0, \quad F_{v,u} = 0. \tag{13}$$

In this formulation, Constraint (8) ensures that $\overline{B}$ lower bound the normalized bandwidth of every node $v \in V$ with positive weight, while Constraint (9) guarantees that bandwidth is not allocated to nodes with weight zero. Constraint (10) is the flow conservation requirement of the relay links. Constraints (11) and (12) guarantee the capacity constraints of the relay links, while Constraint (13) ensures that traffic does not traverse through non-relay links. Unfortunately, this formulation is not a linear problem and we cannot simply use the methods described in Section 4.2.1, since Constraints (10)–(12) depend on the calculated normalized bandwidth $\overline{B}$. However, this formulation becomes a linear program for a fixed $\overline{B}$. This enable us to find the optimal fractional $\overline{B}$ value by performing a binary search over $\overline{B}$ and checking whether there is a fractional flow solution that satisfies the predicted normalized bandwidth $\overline{B}$. We start our search by guessing $\overline{B} = \max_{e \in E} C_e/d_{\max}$, which upper bounds the normalized bandwidth.

Theorem 5. *Consider a graph* $G(V,E)$ *and bounded weight between* $d_{\min}$ *and* $d_{\max}$. *Then,* $\overline{B}^{\mathrm{i}} \geqslant (1+\alpha)\overline{B}^*$, *where* $\alpha = d_{\max}/d_{\min}$.

Proof. Let $\overline{B}^{\mathrm{f}}$ be the normalized bandwidth calculated be the fractional solution. From Theorem 2 and the definition of relay links $R(\overline{B}^{\mathrm{f}})$ follow that the integral flow of any link $e \in R(\overline{B}^{\mathrm{f}})$ satisfy the following expression:

$$\frac{F_e^{\mathrm{i}}}{C_e} \leqslant \frac{F_e^{\mathrm{f}} + d_{\max}\overline{B}^{\mathrm{f}}}{C_e} \leqslant 1 + \frac{d_{\max}}{d_{\min}} \leqslant 1 + \alpha,$$

where F_e^{i} is the integral flow through link e before the scaling. Thus, the over provisioning ratio λ is less than $1+\alpha$. Since, fractional normalized bandwidth upper bounds the optimal integral solution, we get that $\overline{B}^{\mathrm{i}} = \overline{B}^{\mathrm{f}}/\lambda \geqslant \overline{B}^*/(1+\alpha)$ and this completes our proof. □

Corollary 2. *Consider a graph* $G(V \cup \{a\}, E)$ *where all nodes have the same bandwidth demand. Then, the bounded weight scheme is a* 2*-approximation algorithm, i.e.,* $\overline{B}^{\mathrm{i}} \geqslant \frac{\overline{B}^*}{2}$.

4.4. The general scheme

Now we describe our *general algorithm* for arbitrary weights and link capacities. We construct a 5-approximation algorithm by consolidating the *relay group* method used in Section 4.3 and a scaling technique like the one used in [28]. Consider a graph $G(V \cup \{a\}, E)$ and let $d_{\max}$ be the maximal node weight. We divide the nodes with positive weights into disjoint groups D_k, $k > 0$, depending on their weights. A node $v \in V$ with $d_v > 0$ is included in group k if and only if $\frac{d_{\max}}{2^k} < d_v \leqslant \frac{d_{\max}}{2^{k-1}}$. Thus,

$$D_k = \left\{ v | v \in V \wedge \frac{d_{\max}}{2^k} < d_v \leqslant \frac{d_{\max}}{2^{k-1}} \right\}.$$

Recall that the number of sets is bounded by $|V|$, when ignoring empty sets. Our scheme calculates, simultaneously, a fractional solution for all the weight groups that is based on a *relay group* approach. Then, it uses the rounding algorithm described in Section 4.2.2, for obtaining integral flows for each group D_k separately. Consider any integral solution with a normalized bandwidth $\overline{B}$, a link $e \in E$ may serve as a relay for the flow $d_v\overline{B}$ of any node $v \in D_k$ only if $C_e \geqslant d_v\overline{B}$. Thus, for a given normalized bandwidth $\overline{B}$, we define a group of possible relay links, $R_k(\overline{B})$, for each set D_k,

$$R_k(\overline{B}) = \left\{ e | e \in E \wedge C_e \geqslant \frac{d_{\max}}{2^k}\overline{B} \right\}.$$

Next, we formulate a fractional routing problem that allows only links in $R_k(T)$ to forward traffic of flows in D_k. Let denote by $F_{u,v,k}$ the amount of traffic that traverse link (u,v) for serving the nodes in D_k. For generality, let $\overline{B}_{v,k} = \overline{B}_v$ if $v \in D_k$, or 0 otherwise. So, the fractional routing problem is formulated as follows:

$\max \overline{B}$

subject to

$$\forall k > 0,\ \ v \in V \wedge v \in D_k : \overline{B} \leqslant b_{v,k}/d_v, \tag{14}$$

$$\forall k > 0,\ \ v \in V \wedge v \notin D_k : b_{v,k} = 0, \tag{15}$$

$$\forall k > 0,\ \ v \in V : \sum_{u \in N(v) \wedge (u,v) \in R_k(\overline{B})} F_{u,v,k} = b_{v,k} + \sum_{u \in N(v) \wedge (u,v) \in R_k(\overline{B})} F_{v,u,k}, \tag{16}$$

$$\forall e = (u,v) \in E : \sum_{k>0} (F_{u,v,k} + F_{v,u,k}) \leqslant C_e, \tag{17}$$

$$\forall k > 0,\ \ (u,v) \in R_k(\overline{B}) : F_{u,v,k} \geqslant 0, \quad F_{v,u,k} \geqslant 0, \tag{18}$$

$$\forall k > 0,\ \ (u,v) \notin R_k(\overline{B}) : F_{u,v,k} = 0, \quad F_{v,u,k} = 0. \tag{19}$$

In this formulation, Constraint (14) ensures that $\overline{B}$ lower bounds the normalized bandwidth of every node $v \in V$. While, Constraints (15) and (16) are flow conservation requirements for every weight group D_k and every set $R_k(\overline{B})$ of relay links. Constraints (17) and (18) ensure the capacity constraints. Finally, for the sake of completeness, Constraint (17) ensures that a traffic of a node $v \in D_k$ in any weight group does not traverse any link $e \notin R_k(\overline{B})$. Like the formulation of the bounded demand problem, this is not a linear program, but it becomes one for a fixed $\overline{B}$. So, we find the optimal solution by performing a binary search over $\overline{B}$ and checking whether there is a suitable flow assignment that satisfies the predicted normalized bandwidth. Then, we round the flow of each demand group D_k separately and the final solution is the collection of routes of each node $v \in V$.

Theorem 6. *The general scheme is a* 5-*approximation algorithm for the route selection problem for any graph* $G(V \cup \{a\}, E)$ *with arbitrary node weights and link capacities.*

Proof. We denote by j_e the minimal index of any relay group that contains link $e \in E$. In other words, j_e is the minimal index k such that $C_e \geqslant \frac{d_{\max}}{2^k}\overline{B}^{\mathrm{f}}$. According to Theorem 2, for every link $e \in E$ and each weight group D_k follows that $F^{\mathrm{i}}_{e,k} \leqslant F^{\mathrm{f}}_{e,k} + 2\frac{d_{\max}}{2^k}\overline{B}^{\mathrm{f}}$. Where $F^{\mathrm{f}}_{e,k}$ and $F^{\mathrm{i}}_{e,k}$ are the fractional flow and integral flow before the scaling originated by nodes in D_k and traverse through link $e \in E$. Thus,

$$F^{\mathrm{i}}_e \leqslant \sum_{k>0} F^{\mathrm{f}}_{e,k} + 2 \cdot \sum_{k \geqslant j_e} \frac{d_{\max}}{2^k}\overline{B}^{\mathrm{f}} \leqslant F^{\mathrm{f}}_e + 4 \cdot \frac{d_{\max}}{2^{j_e}}\overline{B}^{\mathrm{f}} \leqslant 5C_e.$$

Accordingly, the over provisioning ratio λ is less than 5. Since, fractional normalized bandwidth upper bounds the optimal integral solution, we get that $\overline{B}^{\mathrm{i}} = \overline{B}^{\mathrm{f}}/\lambda \geqslant \overline{B}^{*}/5$ and this completes our proof. □

4.5. User flow routing algorithm

The solution for the user flow model is based on the bounded weight scheme presented in Section

4.3. The algorithm uses similar linear program formulation for calculating the fractional solution when the relay set contains all the link with capacity equal of greater than the predicated normalized bandwidth, i.e.,

$$R(\overline{B}) = \{e | e \in E \wedge C_e \geqslant \overline{B}\}. \quad (20)$$

Unlike the aggregated flow model, we would like to calculate in the rounding phase d_v routes from each node v, where all the routes have the same bandwidth allocation equal to the calculated normalized bandwidth. To this end, we modify the rounding algorithm presented in Section 4.2.2 as follows. Recall that a node weight d_v indicates the number of its associated users. Rather than associated a single token t_v to each node v, we assign d_v tokens to each node $v \in V$, where each token represents a bandwidth allocation equal to the fractional normalized bandwidth $\overline{B}^{\mathrm{f}}$. Beside the initial token assignment, we utilize the same rounding method presented above. Finally, after determining the user individual routes we scale down the normalized bandwidth to meet the capacity constraints. Consequently, from Theorem 2 and by deploying similar arguments to the ones used in the proof of Theorem 3 we conclude,

Corollary 3. *If all the links have the same capacity C then user flow algorithm finds the optimal solution.*

Note that in the case of Corollary 3, the relay set is either empty or contains all links. In addition, by minor modifications of the proof of Theorem 5 it is easy to show,

Corollary 4. *When the links have arbitrary capacities then the user flow algorithm guarantees a 2-approximation ratio.*

5. Simulation results

We evaluate the performance of the proposed scheme via simulations. We compare the performance of the basic Integral solution (INT) described in Section 4.2 with the performance of the Fractional solution (FRAC) and two heuristic algorithms: a plain Shortest-Path (SP) algorithm and a Shortest-Path–Load-Balancing (SP + LB) algorithm. In the SP + LB algorithm, the least-congested shortest path is chosen in a similar way[5] to the scheme presented in [12]. The performance comparison metric is the minimal per-user bandwidth, $\overline{B}$ (i.e., minimal b_v/d_v among all nodes). In the following we present typical results if our simulations.

We consider practical mesh routers (such as BelAir 200) that are associated with 4 backhaul radio interfaces. To simplify the needs for selecting strongly connected mesh topologies with node degrees at most four, we consider a grid-like mesh topologies that coincide squares of sizes 10×10 and 15×15, each one with a single gateway.

We first simulated a case that the gateway node is located at the center of a 10×10 mesh and the users are randomly placed over the topology, while at most one user can be associated with each node (i.e., $d_v = 1$). The capacity of the links between adjacent nodes is set to 10 Mbps (i.e., $C_{u,v} = 10$), while the capacity of the link between the gateway and the infrastructure is set to higher than the sum of the capacity of its links to neighbor nodes. We also assume that the capacity of the access channels is higher that the wireless link capacity between the mesh routers. Fig. 5 presents the simulation result of this case. The X-axis of the graph represents the number of nodes that have an associated user. Each point is obtained through 400 runs and the results of lightly load conditions are not plotted for higher readability. The simulation results indicates that the INT solution performs very close to the FRAC solution (particularly when the network is heavily loaded), and clearly outperforms the heuristic methods. Recall that the FRAC solution is a performance upper bound and is guaranteed to be better than or equal to the optimal integral solution. In our simulations *we did not take the potential gain*

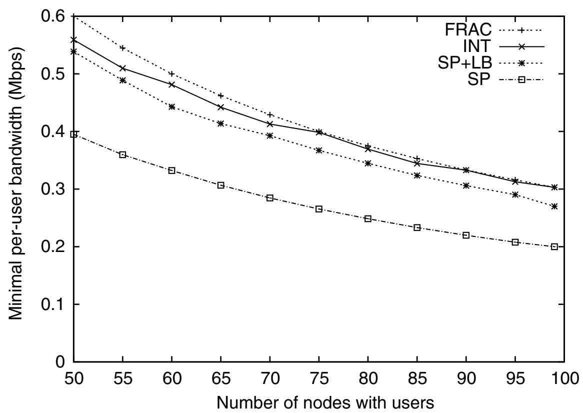

Fig. 5. Minimal per-user bandwidth comparison in a 10×10 mesh with a center-located gateway node ($C_{u,v} = 10$, $d_v = 1$).

[5] Essentially, each edge of the graph maintains a weight representing the current load level of the edge, and a path with a smallest weight sum is chosen.

of using the compression schemes in the INT solution, which may in effect boost the performance of the INT solution over the FRAC (multi-path) solution. We further show in Fig. 7 a comparison of the INT solution with the heuristic methods on the efficiency of the computed routes (i.e., the average length of the routes). The comparison shows that the INT method generates near-optimal routes in terms of the average path length (i.e., very close to the SP method), while the SP + LB method suffers significant extension of the path lengths. That is, the SP + LB method achieves higher performance than the SP method at the expense of longer routes. In contrast the INT solution achieves near-optimal performance without such a downside, which enables it to support delay sensitive application like voice more efficiently.

Fig. 6 plots the simulation results when the gateway node is located near the corner of the grid topology. As compared to the case of the center-located gateway node, the performance gap between the proposed INT solution and other heuristics widens significantly when the gateway node is located near the corner. It is because in latter case the overall routing patterns become more biased and the routing space that the proposed scheme can exploit increases. The average path length comparison (Fig. 8) indicates that a big gap remains between the INT solution and the SP + LB method, while the length of the paths generated by the INT solution is slightly longer than that of the SP method.

Now, we turn to the case that the link capacity and the number of users associated with each node are not fixed. The capacity of each link is randomly

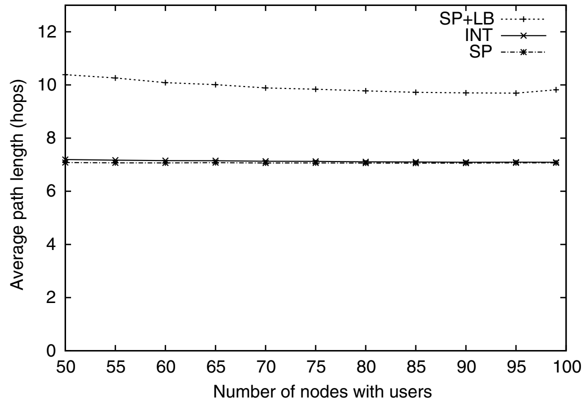

Fig. 7. Average path length comparison in a 10 × 10 mesh with a center-located gateway node ($C_{u,v} = 10$, $d_v = 1$).

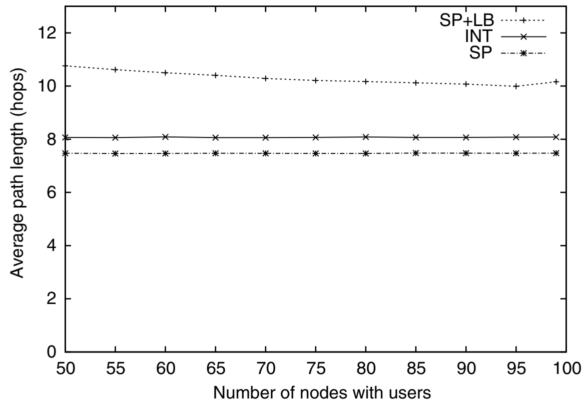

Fig. 8. Average path length comparison in a 10 × 10 mesh with a corner-located gateway node ($C_{u,v} = 10$, $d_v = 1$).

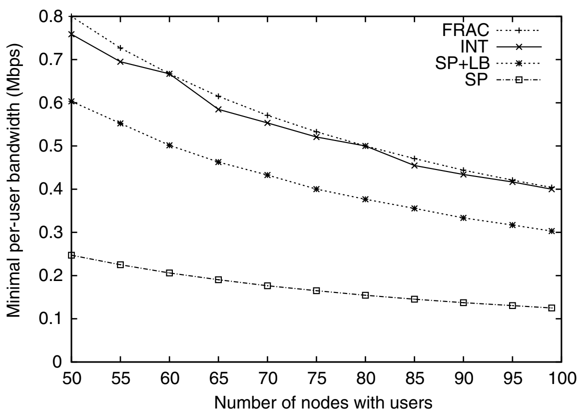

Fig. 6. Minimal per-user bandwidth comparison in a 10 × 10 mesh with a corner-located gateway node ($C_{u,v} = 10$, $d_v = 1$).

chosen between 5 Mbps and 15 Mbps (i.e., $C_{u,v} =$ 5–15). The users are randomly placed so that up to five users are associated with each node (i.e., $d_v = 1$–5). Instead of the more sophisticated methods described in Sections 4.3 and 4.4, we continue to use the basic Integral solution (INT) described in Section 4.2 to demonstrate that even the simple basic method performs fairly well in general cases. Figs. 9 and 10 depict the cases of center-located gateway node and corner-located gateway node, respectively. As expected, the gap between the FRAC solution and the INT solution increases as compared with the case of fixed $C_{u,v}$ and d_v, but other than that the same general trend as in Figs. 5 and 6 can be observed.

To examine the impact of the network size, we also simulate a 15 × 15 mesh. The results are presented in Figs. 11 and 12. As the number of users increases, the rounding error shrinks and the gap

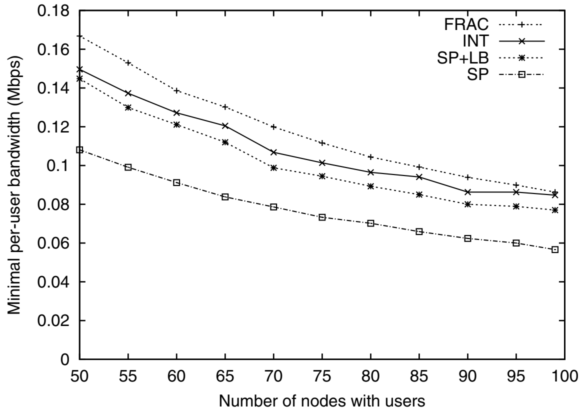

Fig. 9. Minimal per-user bandwidth comparison in a 10×10 mesh with a center-located gateway node ($C_{u,v} = 5–15$, $d_v = 1–5$).

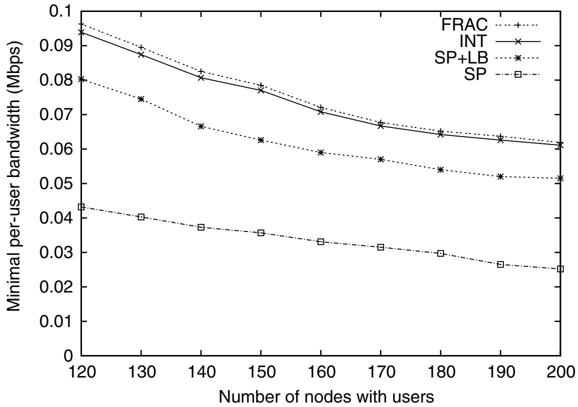

Fig. 11. Minimal per-user bandwidth comparison in a 15×15 mesh with a center-located gateway node ($C_{u,v} = 5–15$, $d_v = 1–5$).

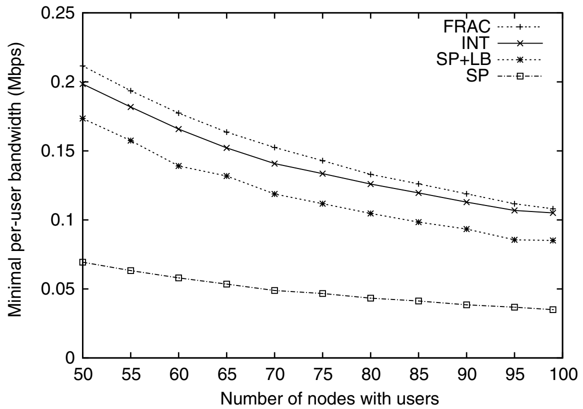

Fig. 10. Minimal per-user bandwidth comparison in a 10×10 mesh with a corner-located gateway node ($C_{u,v} = 5–15$, $d_v = 1–5$).

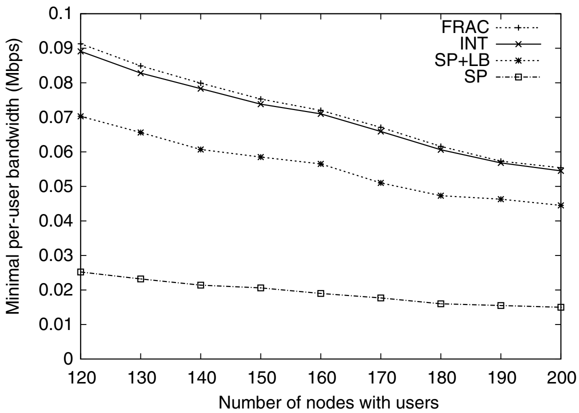

Fig. 12. Minimal per-user bandwidth comparison in a 15×15 mesh with a corner-located gateway node ($C_{u,v} = 5–15$, $d_v = 1–5$).

between the INT solution and the FRAC solution decreases.

Finally, let's examine how much performance gain can be achieved by using the user-flow routing algorithm described in Section 4.5. Since the fractional solution of this algorithm is the same as that of the aggregated flow algorithm, the performance of the user-flow solution must locate between the FRAC solution and the INT solution (for the sake of clarity we omit these curves from the charts). The performance gap between the FRAC and INT solutions is very small in all simulations cases presented, the performance gain by the user-flow method is minimal. To conclude, our simulation results reconfirm the argument made by Ganjali and Keshavarzian in [17] that the performance gain of multi-path routing is not big.

6. Conclusion

In this paper, we focused on determining the routes and bandwidth allocation of the traffic flows of WMNs for maximizing the fair share allocated to the users. We presented polynomial-time algorithms for this purpose and analyzed the quality of our solutions against the optimal solutions. The simulations show that our algorithms, indeed, find near-optimal solutions. These algorithms can be used under the centralized management architecture, in which a NOC located in the wired network performs network management functions for WMNs. The basic idea of our algorithms is not limited to the current LP problem formulations, but can rather be applied to various other problem formulations, for instance, ones with different interference models.

References

[1] R. Bruno, M. Conti, E. Gregori, Mesh networks: commodity multihop ad hoc networks, IEEE Commun. Mag. 47 (3) (2005) 123–131.

[2] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: Proceedings of ACM Mobicom'04, Philadelphia, PA, USA, September 2004, pp. 114–128.

[3] A. Raniwala, T.-C. Chiueh, Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. in: Proceedings of IEEE INFOCOM'05, Miami, FL, USA, March 2005.

[4] R. Karrer, A. Sabharwal, E. Knightly, Enabling large-scale wireless broadband: the case for TAPs, in: Proceedings of HotNets 2003, Cambridge, MA, USA, November 2003.

[5] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks 47 (4) (2005) 445–487.

[6] S. Ramanathan, M. Steenstrup, A survey of routing techniques for mobile communications networks, Journal of Mobile Networks and Applications (MONET) 1 (2) (1996) 89–104.

[7] D. De Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, in: Proceedings of ACM Mobicom'03, San Diego, CA, USA, September 2003, pp. 134–146.

[8] J. Tang, G. Xue, W. Zhang, Interference-aware topology control and QoS routing in multi-channel wireless mesh networks, in: Proceedings of ACM MobiHoc'05, Urbana-Champaign, IL, USA, May 2005, pp. 68–76.

[9] T.S. Rappaport, Wireless Communications: Principle and Practice, second ed., Prentice Hall, 2002.

[10] P. Kyasanur, N. Vaidya, Routing and interface assignment in multi-channel multi-interface wireless networks, in: Proceedings of IEEE WCNC'05, 2005.

[11] P.-H. Hsiao, A. Hwang, H.T. Kung, D. Vlah, Load balancing routing for wireless access networks, in: Proceedings of IEEE INFOCOM'01, Anchorage, Alaska, USA, 2001, pp. 986–995.

[12] J. He, J. Chen, S.-H. Gary Chan, Extending WLAN coverage using infrastructureless access points, in: Proceedings of HPSR 2005, May 2005.

[13] K. Jain, J. Padhye, V. Padmanabhan, L. Qiu, Impact of interference on multi-hop wireless network performance, in: Proceedings of ACM Mobicom'03, San Diego, CA, USA, September 2003, pp. 66–80.

[14] M. Kodialam, T. Nandagopal, Characterizing the achievable rates in multihop wireless networks, in: Proceedings of ACM Mobicom'03, San Diego, CA, USA, September 2003, pp. 42–54.

[15] M. Kodialam, T. Nandagopal, Achievable rates in multi-hop wireless mesh networks with orthogonal channels, IEEE/ACM Transactions of Networking, September 2005, in preparation.

[16] P. Pham, S. Perreau, Performance analysis of reactive shortest single-path and multi-path routing mechanism with load balance, in: Proceedings of IEEE INFOCOM'03, Sun Francisco, CA, USA, 2003.

[17] Y. Ganjali, A. Keshavarzian, Load balancing in ad hoc networks: single-path routing vs. multi-path routing, in: Proceedings of IEEE INFOCOM'03, Hong Kong, 2004.

[18] O.C. Ibe, Converged Network Architectures, Delivering Voce and Data over IP, ATM and Frame Relay, Wiley, 2002.

[19] T.V. Lakshman, T. Nandagopal, R. Ramjee, K. Sabnani, T. Woo, The SoftRouter architecture, in: Proceedings of ACM ACM HotNets-III, San Diego, CA, USA, November 2004.

[20] C. Bormann, et al., RObust Header Compression (ROHC): framework and four profiles: RTP, UDP, ESP, and uncompressed (RFC 3095), July 2000.

[21] V. Gambiroza, B. Sadeghi, E. Knightly, End-to-end performance and fairness in multihop wireless backhaul networks, in: Proceedings of ACM Mobicom'04, Philadelphia, PA, USA, September 2004, pp. 287–301.

[22] Y. Dinitz, N. Garg, M.X. Goemans, On the single-source unsplittable flow problem, in: IEEE Symposium on Foundations of Computer Science, FOCS'98, 1998, pp. 290–299.

[23] D.E. McDysan, D.L. Spohn, ATM Theory and Application, McGraw-Hill, 1998.

[24] B.S. Davie, Y. Rekhter, MPLS: Technology and Applications, Morgan Kaufmann Publishers, 2000.

[25] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, Freeman Publication, New York, 1979.

[26] G. Zussman, A. Segall, Energy efficient routing in ad hoc disaster recovery networks, in: Proceedings of IEEE INFOCOM'03, Sun Francisco, CA, USA, 2003.

[27] N.Garg, J. Kemann, Faster and simpler algorithms for multicommodity flow and other fractional packing problems, in: Proceedings of the 39th Annual IEEE Computer Society Conference on Foundations of Computer Science, FOCS'98, Palo Alto, CA, USA, November 1998, pp. 300–309.

[28] S.G. Kolliopoulos, C. Stein, Improved approximation algorithms for unsplittable flow problems, in: Proceedings of the 38th Annual IEEE Computer Society Conference on Foundations of Computer Science, FOCS'97, Miami Beach, FL, USA, October 1997, pp. 426–435.

Yigal Bejerano received the B.Sc. degree in computer engineering (summa cum laude), the M.Sc. degree in computer science, and the Ph.D. degree in electrical engineering from the Technion – Israel Institute of Technology, Haifa, Israel, in 1991, 1995, and 2000, respectively. He is currently a Member of the Technical Staff with the Networking Research Center, Bell Laboratories, Lucent Technologies, Murray Hill, NJ. His research interests are mainly management aspects of high-speed and wireless networks, including the areas of mobility management, network monitoring, topology discovery, quality of service (QoS) routing, wireless LAN, and wireless mesh networks.

Seung-Jae Han received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, Korea, and the Ph.D. degree in Computer Science and Engineering from the University of Michigan, Ann Arbor. He was a Member of Technical Staff of the Wireless Research Laboratory, Bell Laboratories, Lucent Technologies, Murray Hill, NJ. Currently he is an associate professor in the computer science department of Yonsei University, Seoul, Korea. His research interests include mobile networking, wireless Internet, and network management.

Amit Kumar received the B.S. degree from Indian Institute of Technology, Kanpur, India, and the Ph.D. degree from Cornell University. He was a Member of Technical Staff in Bell Laboratories, Lucent Technologies, Murray Hill, NJ. Currently he is an assistant professor in the computer science department of Indian Institute of Technology, New Delhi, India.