A Channel and Rate Assignment Algorithm and a Layer-2.5 Forwarding Paradigm for Multi-Radio Wireless Mesh Networks

Stefano Avallone, *Associate Member, IEEE*, Ian F. Akyildiz, *Fellow, IEEE*, and Giorgio Ventre, *Member, IEEE*

***Abstract*—The availability of cost-effective wireless network interface cards makes it practical to design network devices with multiple radios which can be exploited to simultaneously transmit/receive over different frequency channels. It has been shown that using multiple radios per node increases the throughput of multi-hop wireless mesh networks. However, multi-radios create several research challenges. A fundamental problem is the joint channel assignment and routing problem, i.e., how the channels can be assigned to radios and how a set of flow rates can be determined for every network link in order to achieve an anticipated objective. This joint problem is NP-complete. Thus, an approximate solution is developed by solving the channel assignment and the routing problems separately. The channel assignment problem turns out to be the problem to assign channels such that a given set of flow rates are schedulable and itself is shown to be also NP-complete. This paper shows that not only the channels but also the transmission rates of the links have to be properly selected to make a given set of flow rates schedulable. Thus, a greedy heuristic for the *channel and rate assignment* problem is developed. Algorithms to schedule the resulting set of flow rates have been proposed in the literature, which require synchronization among nodes and hence modified coordination functions. Unlike previous work, in this paper a forwarding paradigm is developed to achieve the resulting set of flow rates while using a standard MAC. A bi-dimensional Markov chain model of the proposed forwarding paradigm is presented to analyze its behavior. Thorough performance studies are conducted to: a) compare the proposed greedy heuristic to other channel assignment algorithms; b) analyze the behavior of the forwarding paradigm through numerical simulations based on the Markov chain model; c) simulate the operations of the forwarding paradigm and evaluate the achieved network throughput.**

***Index Terms*—Multi-radio wireless mesh networks, channel assignment, physical model of interference, layer-2.5 forwarding paradigm.**

I. Introduction

Wireless MESH NETWORKs (WMNs) are comprised of a backbone of mesh routers which collect and relay the traffic generated by mesh clients [1]. Mesh routers have limited (if any) mobility and are usually connected through wireless links. If equipped with the necessary gateway and bridging functionalities, mesh routers enable the integration of wireless mesh networks with other networks such as the Internet, cellular, IEEE 802.15, IEEE 802.16, sensor networks, etc. Mesh clients are typically mobile and rely on mesh routers to deliver data to the intended destinations.

Wireless mesh networks are attractive for several applications, e.g., wireless last mile access of ISPs, wireless enterprise backbone networks, building automation, broadband home networking, community, neighborhood networks, etc. The main reason is the low cost of deployment and maintenance due to the absence of a wired infrastructure. However, the use of wireless links causes communications among routers to suffer from environmental noise and interference problems. WMNs, being multi-hop networks, are particularly affected by such problems, as both adjacent hops on the same path and neighboring paths can cause interference.

Interference can be alleviated if different node pairs in a neighborhood use non-interfering frequency channels. In this regard, 3 and 12 non-overlapping channels are defined by the IEEE 802.11b/g and IEEE 802.11a standards, respectively. In case network nodes are equipped with a single radio, the use of multiple channels in the network leads to disconnected subsets of nodes, as each node can only communicate with the neighbor nodes using the same channel. To provide connectivity, new MAC protocols have been developed, e.g., in [2], [3], which enable nodes to switch their radio to a different channel when needed. However, the channel switching requires fine-grained synchronization among nodes in order to avoid the *deafness problem*, i.e., the transmitter and the intended receiver may be on different channels. Also the time for channel switching which can be in the range of a few milliseconds to a few hundred microseconds [4] may be unacceptable for most real-time multimedia applications.

Recently, given the availability of low cost wireless devices, a different solution for the problem of reducing the interference is being proposed, which consists in endowing each node with multiple radios. Radios are set to different channels and no channel switching is required. Thus, each node can simultaneously communicate over different channels, which has been shown to reduce the interference and increase the network throughput [5], [6].

Challenging research issues in multi-radio wireless mesh networks are the channel assignment and the routing problems, i.e., the problems to find, respectively, an assignment of channels to

Manuscript received February 08, 2007; revised August 23, 2007; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor E. Knightly. First published March 12, 2008; current version published February 19, 2009. This work was supported in part by the Italian Ministry for Education, University and Research (MIUR) in the framework of the NADIR Project (PRIN Program).

S. Avallone is with the Department of Computer Engineering, University of Naples, 80125 Naples, Italy (e-mail: stavallo@unina.it).

I. F. Akyildiz is with the Broadband Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: ian@ece.gatech.edu).

G. Ventre is with the Department of Computer Engineering, University of Naples, 80125 Naples, Italy (e-mail: giorgio@unina.it).

Digital Object Identifier 10.1109/TNET.2008.918091

radios and a set of flow rates for every network link which optimize a given objective. Common optimization objectives are to maximize the aggregate network throughput, to verify the achievability of a traffic demand vector or to maximize the minimum end-to-end rate.

Channel assignment and routing are not independent problems, as solving one problem requires a solution for the other problem. Indeed, solving the routing problem requires the knowledge of the bandwidth available on all the links. However, we do not have this knowledge prior to solving the channel assignment problem. This is because the channel assignment algorithm determines the sets of links sharing the same channel and accordingly their available bandwidth. Likewise, the channel assignment algorithm needs to be aware of the flow rate expected on the network links. This information enables the algorithm to assign channels such that the available bandwidth on each link exceeds the required flow rate. However, this information can only be determined by solving the routing problem. Hence, it is clear that the channel assignment and the routing problems are closely inter-dependent on each other and must be jointly solved. Unfortunately, the joint channel assignment and routing problem is NP-complete [7].

An approximate algorithm for the channel assignment and routing problem typically consists of the following three steps:

1) **Determine the pre-computed flow rates:** A pre-computed flow rate is determined for every link based on the given optimization objective.
2) **Determine the channel assignment:** Channels are assigned to radios in the attempt to make the pre-computed flow rates returned by the previous step schedulable, i.e., actually achievable considering the interference among transmissions over the same channel.
3) **Adjust the pre-computed flow rates:** The pre-computed flow rates returned by the first step may be adjusted in order to obtain a set of schedulable flow rates given the computed channel assignment.

In this paper we develop a centralized heuristic for the joint channel assignment and routing problem. First, we propose a method to determine a set of pre-computed flow rates with the objective of maximizing the aggregate network throughput. Then, we formally define the problem to find, if any exists, a channel assignment such that the given set of pre-computed flow rates are schedulable and show it is NP-complete. Then, we show that the physical transmission rate of the network links also affects the possibility to make the set of pre-computed flow rates schedulable. Hence we exploit the availability of multiple transmission rates ensuing from different modulation schemes and tackle a *channel and rate assignment* problem. To our knowledge, the channel assignment problem has not been jointly studied with the problem to select the transmission rates so far. We develop a greedy heuristic having the property that the returned channel assignment is invariant for scaling of the pre-computed flow rates.

Given the computed channel assignment, the pre-computed flow rates may need to be adjusted to provide a set of schedulable flow rates. Previous work has shown how to build an interference-free scheduling, which, however, requires synchronization among nodes and hence modified coordination functions. In this paper, we stick to standard MAC protocols and develop a novel *Layer-2.5* forwarding paradigm which requires no modification to the standard coordination functions. The proposed forwarding paradigm enables each mesh router to autonomously take forwarding decisions in such a way that the average transmission rate on each link approximates the computed flow rate. Our forwarding paradigm also addresses the need to react to link failures/quality degradations, which may frequently occur in wireless networks. Our solution abandons the use of routing tables. However, some information on the network topology is still required for its operations. Hence the term *Layer-2.5* we use to refer to our proposed paradigm. To our knowledge, no other distributed forwarding/routing protocols have been presented aiming to enforce a given set of flow rates in a multi-radio wireless mesh network. We also present a bi-dimensional Markov chain model of the proposed Layer-2.5 forwarding paradigm to study how its behavior is affected by the settings of some parameters.

The rest of the paper is structured as follows. In Section II, we give an overview of the related work. In Section III, we formalize the channel and rate assignment problem. In Section IV, we present all the three steps of our centralized heuristic. In Section V, we illustrate the operation and the model of the proposed Layer-2.5 forwarding paradigm. In the next section we present simulation experiments to evaluate the performance of our scheme and compare it to other existing schemes. Finally, in Section VII we conclude the paper.

II. Related Work

The channel assignment problem in multi-radio WMNs has been investigated in the literature recently. Many proposals aim to minimize some network-wide measure of interference and do not study the channel assignment problem in conjunction with the routing problem. For instance, a centralized channel assignment algorithm is presented in [8] which aims to limit interference while preserving connectivity. Given a K-connected potential communication graph, the goal is to find a channel assignment which minimizes the maximum among the size of the collision domain of all the links subject to the constraint that the induced graph must still be K-connected. A polynomial time recursive heuristic based on the use of a conflict graph is proposed in [9]. Each network link is associated with a *link conflict weight*. Two objectives are considered which lead to two different variants: GreedyMax, that aims to minimize the maximum link conflict weight at any link, and GreedyAvg, that aims to minimize the average link conflict weight over all the links. A centralized channel assignment algorithm is presented in [10] which takes the traffic generated by mesh clients into account. Each mesh router periodically captures packets generated by the mesh clients and measures the number of senders and per second utilization for each channel. Then, it ranks each channel based on the number of clients and the channel utilization. Each link is assigned the highest ranked channel that does not conflict with the channel assignment of its neighbors. A distributed channel assignment algorithm together with a distributed routing protocol are proposed in [6]. At any time, each node joins the neighbor which minimizes the cost to reach a gateway and sends all the packets destined to the wired network to such neighbor. Joining a new neighbor requires to update the routing tables of all the nodes along the paths to the previous and the new gateways. Each node only assigns channels to the radios used to communicate with its children nodes. Channels are selected based on their usage by the interfering nodes.

Other proposals study the joint channel assignment and routing problem. An iterative routing algorithm based on traffic profiles is proposed in [5]. Given the set of initial link flow rates, channels are assigned in the attempt to have the resulting available bandwidth on each link exceed the link flow rate. The available bandwitdh values are estimated as a fraction of the link capacity and are used as input to the routing algorithm, which computes the shortest feasible path for every flow of the given traffic profile. The resulting flow allocated on each link is used as link flow rate for the next iteration, in which a new channel assignment is computed. In [7] an approximate solution for the joint channel assignment and routing problem is developed which optimizes the network throughput subject to fairness constraints. The traffic load that each mesh router collects from its clients and has to route towards the mesh gateways is assumed to be known. An ILP (Integer Linear Program) is formulated to find the link flow rates and the channel assignment that maximize the fraction (the same for all the nodes) of the traffic load that each node gets delivered to the wired network. Solving the LP relaxation provides a possibly unfeasible channel assignment. The channel assignment algorithm aims to fix this unfeasibility. The flow on the graph is then readjusted and scaled to ensure a feasible channel assignment and routing. Also a scheduling algorithm is proposed in [7] to produce an interference free link schedule. The problem how to verify the feasibility of a given set of flows between source-destination pairs is investigated in [11]. The goal is to determine the maximum scaling factor for the flows that still satisfies the constraints on the number of radios per node and the schedulability constraint. The binary variables in these constraints are approximated by appropriate continuous variables. The resulting LP is solved by using a primal-dual approach based on shortest path routing. The solution provides a possibly unfeasible channel assignment and a set of link flow rates. A greedy channel assignment algorithm and a subsequent flow scaling are then used to ensure a feasible channel assignment and routing. Unlike this previous work, our proposal does not require the knowledge of the traffic demands. Also, most importantly, our proposal addresses the channel and rate assignment problem, which provides both a channel and a transmission rate for each link.

The joint channel assignment and congestion control problem is studied in [12] based on [13]. TCP congestion control mechanism is analyzed in [14] as an approximate distributed algorithm solving a network utility maximization problem. This analysis is used in [13] for multihop wireless networks. In particular, the Shannon's formula is used to model the capacity of wireless links and the optimal source rates and transmission powers are determined which maximize the network utility. In [12] instead multi-radio nodes with fixed transmission powers are considered and the optimal source rates and channels are calculated such that the network utility will be maximized. Also the work

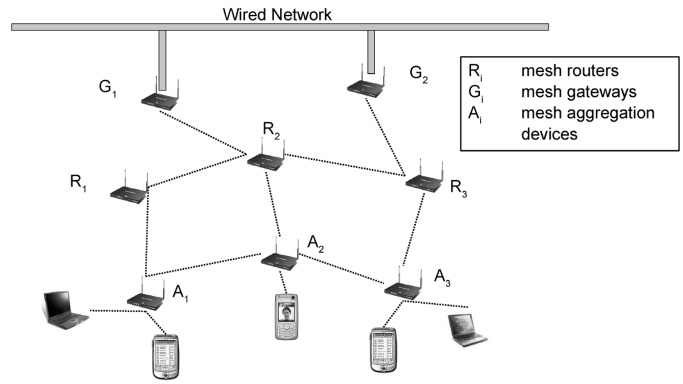

Fig. 1. Wireless mesh network reference architecture.

in [12] does not consider the effect of using different transmission rates ensuing from different modulation schemes.

The problem to select a transmission rate from among a set of available ones has been investigated in the context of rate adaptation (e.g., [15] and references therein). The work in this area aims to select a transmission rate for a radio on a per-packet basis depending on local conditions. Our centralized algorithm instead selects a transmission rate for a link taking the network-wide effect of such a choice into account. Also, this choice is not made on a per-packet basis.

III. System Model and Problem Formulation

We consider the WMN architecture given in Fig. 1. Some of the mesh routers denoted as *mesh aggregation devices* collect user traffic and forward it to the wired network through multiple hops across the WMN. Mesh routers connected to the wired network are denoted as *mesh gateways*. We assume that each mesh router u is equipped with $k(u) \geq 1$ radio interfaces and there are $|\mathcal{C}|$ available channels. For every radio, we assume a fixed transmission power, while the transmission rate can be selected in the (increasingly) ordered set $\{r_m\}_{m=1}^{M}$.

The impact of the interference can be formally accounted for through either of the interference models defined in [16]: the *protocol* model that assumes interference to be an all-or-nothing phenomenon and the *physical* model that considers a transmission successful if the Signal-to-Interference and Noise Ratio (SINR) at the receiver is sufficiently high to decode the signal. In this paper we consider the latter model and define the SINR at receiver v when a signal is transmitted by u as

$$\mathrm{SINR}_{uv} = \frac{G_{uv}P_u}{\sum_{x \rightarrow y \neq u \rightarrow v} G_{xv}P_x + n_v}$$

where P_u is the transmission power emitted by u, G_{uv} is the gain of the radio channel between u and v, and n_v is the thermal noise at receiver v. If u transmits at rate r_m, the receiver v can correctly decode the signal if $\mathrm{SINR}_{uv} \geq \gamma_{r_m}$, where γ_{r_m} denotes the minimum SINR required to correctly decode a signal modulated using the scheme associated with the rate r_m. It is a known result that the higher the transmission rate, the higher is the SINR threshold.

We model the WMN as a directed graph $G_I = (V, E_I)$, where V is a set of nodes each representing a mesh router. Given two nodes $u, v \in V$, the directed edge $u \to v \in E_I$ iff

$$\frac{G_{uv}P_u}{n_v} \geq \gamma_{r_M} \tag{1}$$

i.e., in the absence of transmissions on other links, the signal to noise ratio is larger than the SINR threshold for the highest available transmission rate r_M.

We refer to G_I as the *potential communication graph*, as an edge $u \to v \in E_I$ indicates that u can transmit to v provided that they are assigned a common channel. Bidirectional transmissions are required to acknowledge the receipt of packets and hence we assume that transmission powers are set such that $u \to v \in E_I \Rightarrow v \to u \in E_I$. The capacity $c(u \to v)$ of the directed edge $u \to v$ is the transmission rate r_m used by u to transmit to v. We denote by $V_A \subseteq V$ and $V_G \subseteq V$ the set of mesh aggregation devices and mesh gateways, respectively.

A link $x \to y \in E_I$ is said to *potentially interfere* with $u \to v \in E_I$ if a simultaneous transmission on $x \to y$ prevents v from correctly decoding the signal from u. We denote the *potential* collision domain of a link $u \to v$, i.e., the set of all the links that potentially interfere with it, by

$$\mathcal{N}(u \to v) = \left\{ x \to y \in E_I | \frac{G_{uv}P_u}{G_{xv}P_x + n_v} < \gamma_{c(u \to v)} \right\}.$$

In other words, none of the links in $\mathcal{N}(u \to v)$ can be active at the same time as $u \to v$. The term "potentially" is used to underline that such links may interfere with $u \to v$ only if they use the same channel. We note that such definition implies that: (i) the potential collision domain of a link depends on the transmission rate selected for that link; (ii) $u \to v \in \mathcal{N}(u \to v)$; (iii) $x \to y \in \mathcal{N}(u \to v) \nRightarrow u \to v \in \mathcal{N}(x \to y)$.

A channel assignment $\mathcal{A}$ assigns a set $\mathcal{A}(u)$ of channels $(|\mathcal{A}(u)| \leq k(u))$ to each node $u \in V$. Thus, $\mathcal{A}$ induces a new graph model $G = (V, E)$ where two nodes u and v are connected if (1) holds (thus $u \to v \in E_I$) *and* they share at least one common channel, i.e., $\mathcal{A}(u) \cap \mathcal{A}(v) \neq \emptyset$. In case u and v share multiple channels, the set E may include as many links between the two nodes as the number of common channels. To differentiate among those links and stress that a link has been assigned channel c, we use the notation $u \xrightarrow{c} v$. We note that a link in E_I does not have any corresponding link in E if the end nodes do not share any channel.

If $x \to y \in \mathcal{N}(u \to v)$ and x, y, u and v are assigned a common channel c, then $x \xrightarrow{c} y \in E$ is said to interfere with $u \xrightarrow{c} v \in E$. Also, the collision domain of a link $u \xrightarrow{c} v \in E$ can be determined as the subset of the potential collision domain of $u \to v$ including all the links that are assigned channel c, i.e.,

$$\mathcal{D}_{\text{coll}}(u \xrightarrow{c} v) \stackrel{def}{=} \left\{ x \xrightarrow{c} y \in E | x \to y \in \mathcal{N}(u \to v) \right\}.$$

We note that $v \xrightarrow{c} u$ *must* belong to $\mathcal{D}_{\text{coll}}(u \xrightarrow{c} v)$ as a single radio cannot transmit and receive simultaneously.

As described earlier, the first step of an approximate solution for the channel assignment and routing problem typically provides a pre-computed flow rate f for every link $u \to v \in E_I$. The channel assignment problem is to find, if any exists, a channel assignment such that the given set of pre-computed flow rates are schedulable. We note that $f(u \to v)$ may be split over multiple $u \xrightarrow{c} v$ links of G in case u and v share multiple channels. We aim to establish a *sufficient* (but not necessary) condition for a set of pre-computed flow rates to be schedulable and then find, if any exists, a channel assignment that satisfies such condition (though the lack of a channel assignment that satisfies the sufficient condition does not imply that the set of pre-computed flow rates are not schedulable).

To establish a sufficient condition for a set of pre-computed flow rates to be schedulable, we recall that by definition a link not included in $\mathcal{D}_{\text{coll}}(u \xrightarrow{c} v)$ does not interfere with $u \xrightarrow{c} v$. If only one link per collision domain is active in every time slot, then the corresponding transmissions do not interfere with each other (here we neglect the cumulative effect of different transmissions). An interference-free schedule can thus be built by avoiding simultaneous transmissions on links belonging to the same collision domain. Hence, the period T of the schedule has to be sufficiently long to allow each link in every collision domain to carry the required amount of data. In every time period T, each link e_0 has to carry an amount of data equal to $f(e_0)T$. Since the transmission of such amount of data at a rate $c(e_0)$ takes $\frac{f(e_0)}{c(e_0)}T$ time, simultaneous transmissions on the links belonging to the collision domain of the generic link e can be avoided if $\sum_{e_0 \in \mathcal{D}_{\text{coll}}(e)} \frac{f(e_0)}{c(e_0)}T \leq T$. Therefore

$$\sum_{e_0 \in \mathcal{D}_{\text{coll}}(e)} \frac{f(e_0)}{c(e_0)} \leq 1 \quad \forall e \in E \tag{2}$$

is a sufficient condition for a given set of pre-computed flow rates to be schedulable. A proof by construction is given in [7], where an algorithm that finds an interference-free scheduling in case the above sufficient condition holds is presented.

The channel assignment affects the satisfaction of the above sufficient condition as it determines the composition of the various collision domains in the network. Hence we consider the problem to find, if any, a channel assignment that satisfies the sufficient condition for a given set of pre-computed flow rates to be schedulable:

Channel assignment problem (decision version): *Given the potential communication graph $G_I(V, E_I)$ representing a WMN and a set of pre-computed flow rates $\{f(e)\}_{e \in E_I}$, determine whether there exists a channel assignment such that:*

$$\max_{e \in E} \sum_{e_0 \in \mathcal{D}_{\text{coll}}(e)} \frac{f(e_0)}{c(e_0)} \leq 1$$

Such a decision problem is NP-complete (a proof can be found in the Appendix). Hence it is unlikely that a polynomial time algorithm exists to determine whether a channel assignment satisfying the above inequality can be found. Therefore, heuristics are typically proposed that first determine a channel assignment and then adjust the pre-computed flow rates to obtain a set of schedulable flow rates given the computed channel assignment. Thus, the returned set of schedulable flow rates is likely to be different than the given set of pre-computed flow rates. Our goal is to find a channel assignment and a corresponding set of schedulable flow rates that are as *close* as possible to the given set of pre-computed flow rates. In particular, we seek a channel assignment which minimizes the scaling factor $\lambda \geq 1$, where $\{\frac{f(e)}{\lambda}\}_{e\in E}$ is a set of flow rates satisfying the sufficient condition for schedulability given the computed channel assignment.

When the channel assignment is determined, for every link $e \in E$ we can compute $\sum_{e_0\in\mathcal{D}_{\rm coll}(e)} \frac{f(e_0)}{c(e_0)}$, which for conciseness we refer to as the *total utilization* $U_{\rm tot}(e)$ of the collision domain of link e. The maximum among the total utilization of all the collision domains turns out to be the minimum scaling factor which yields a set of flow rates satisfying the sufficient condition (2). It suffices to observe that, if all the pre-computed flow rates are divided by a value α, then the total utilization of every collision domain becomes α times smaller. Therefore, we tackle the problem to find a channel assignment that minimizes $\max_{e\in E} U_{\rm tot}(e)$. This problem turns out to be the optimization version of the decision problem stated above, which is NP-complete. Thus, it is not practical to compute an optimal solution and hence we develop a heuristic algorithm described in Section IV-B.

The total utilization $U_{\rm tot}(e) = \sum_{e_0\in\mathcal{D}_{\rm coll}(e)} \frac{f(e_0)}{c(e_0)}$ of the collision domain of link e is also affected by the capacity of all the links in that collision domain. At a first glance, we may conclude that we only have to select the highest transmission rate r_M for all the links in order to minimize the total utilization of the collision domain. However, decreasing the transmission rate on link e brings with it a lower SINR threshold, which means the transmission on more links may be compatible with the transmission on e. In general, $\mathcal{D}_{\rm coll}(e|c(e) = r_i) \subseteq \mathcal{D}_{\rm coll}(e|c(e) = r_j)$ for $i < j$. Thus, decreasing the transmission rate on link e may help reduce the total utilization of its collision domain. Therefore, we consider a *joint channel and rate assignment problem*, i.e., the problem to select a channel and a transmission rate for every network link in order to minimize $\max_{e\in E} U_{\rm tot}(e)$. Finally, we note that decreasing the transmission rate on a link e has no effect on the composition of the collision domain of the other links (since the transmission power does not change). However, it affects the total utilization of the other collision domains since the ratio $\frac{f(e)}{c(e)}$ increases.

IV. An Approximate Solution for the Channel and Rate Assignment and Routing Problem

This section presents the approximate algorithm we develop for the joint channel and rate assignment and routing problem. Each of the following subsections details one of the three steps our algorithm is composed of.

A. Determine the Pre-Computed Flow Rates

Channel assignment algorithms need an estimate of the link flow rates that optimize a given objective. Prior to presenting our solution, we review some methods proposed in related work to determine the pre-computed flow rates:

- As described in Section II, in [7] and [11] the joint channel assignment and routing problem is formulated as an integer linear program (ILP). Then, the linear relaxation of such ILP is computed, which provides a possibly unfeasible solution for the joint problem. Nonetheless, the flow rates returned by solving the relaxed program may be used as the pre-computed flow rates which feed the channel assignment algorithm.
- The optimization objective considered in [5] is to maximize the cross-section goodput over all the source-destination pairs in the network. The estimated traffic demand between each such pair is assumed to be known. Determining the pre-computed flow rates requires to find all the acceptable paths between each source-destination pair. Assuming a perfect load balancing, each pair contributes to the pre-computed flow rate of a link an amount of flow equal to a given fraction of the corresponding traffic demand. This fraction equals the ratio of the number of acceptable paths that pass that link to the total number of acceptable paths between the source-destination pair.

The optimization objective we consider in this paper is to maximize the network aggregate throughput. We point out that our solution does not require an estimation of the traffic demands. We are only given the potential communication graph, as the channel assignment is computed in a subsequent step. Hence, we can maximize the throughput either assuming one single channel is available or assuming that interference does not arise. The former assumption leads to underestimate the optimal flow rates, as it does not account for the possibility to simultaneously transmit over links that are assigned different channels. The latter assumption leads to overestimate the optimal flow rates, as the effect of the interference is to prevent simultaneous transmissions over interfering links. We opt for the latter assumption as it allows for a simple way to compute the maximum throughput, while the problem to find the maximum achievable throughput of the potential communication graph under the protocol interference model is NP-complete [17]. Also, given that the channel assignment returned by our algorithm is invariant for scaling of the pre-computed flow rates (Section IV-C), the main objective of a method to determine the pre-computed flow rates is to identify the relative importance of links in carrying traffic rather than accurately determine absolute values.

The method we propose to determine a set of pre-computed flow rates is based on the observation that mesh routers have to forward packets towards the wired network, regardless of which particular gateway is used. Thus, mesh aggregation devices collecting user traffic do not have to forward each packet to a specific mesh gateway, but can direct it to *any* of the mesh gateways. Thus, we consider an extra node, referred to as the *supersink*, connected to the gateways by links of infinite capacity and compute the maximum network flow [18] between each aggregation device and the supersink. Each maximum flow computation associates each link with an amount of flow. For each link, the pre-computed flow rate is then given by the sum of such amount of flow over all the maximum flow computations.

B. A Greedy Heuristic for the Channel and Rate Assignment Problem

In this subsection we present the FCRA (Flow-based Channel and Rate Assignment) algorithm, a greedy heuristic for the channel and rate assignment problem defined in Section III. The proposed heuristic ensures network connectivity by including (at least) one link in E for every link in E_I. We illustrate the basic concepts of the FCRA algorithm, detail its operation through a pseudocode and analyze its complexity.

The FCRA algorithm (Fig. 2) adopts a greedy strategy. All the links of the potential communication graph are inserted into a priority queue Q and extracted one-by-one to originate one or more links to be included in E. The channel and the rate assigned to such links are selected to minimize the maximum total utilization at the time they are established. A channel can be assigned to a link only if one radio on both the end nodes can be set to that channel. In order to ensure that all the extracted links share a common channel, we initialize one radio of every node to a fixed channel, say it channel 1. Thus, a temporary link on channel 1 exists between every two nodes connected in the potential communication graph and is initially included in E. Each such link is also initially assigned the highest available transmission rate (line 1–3).

Links are extracted from Q in decreasing order of priority, where the priority of a link $u \rightarrow v$ is the total utilization of the collision domain of the corresponding temporary link $u \xrightarrow{c_T} v$ (c_T is channel 1 unless modified by the optimization step). When a link $u \rightarrow v$ is extracted, the goal is to determine the set of links $u \xrightarrow{c} v$ that will replace the temporary link $u \xrightarrow{c_T} v$ in E. Links can be clearly established on channels in common to both end nodes. We denote by $\mathcal{S}_C$ the set of channels shared by the end nodes. Additionally, we allow at most one radio on each end node to be assigned a new channel. Thus, the set $\mathcal{S}$ of candidate channels for the establishment of links $u \xrightarrow{c} v$ includes $\mathcal{S}_C$ and is determined as follows (lines 11–17). If both the end nodes of the extracted link have available radios (i.e., radios with no channel assigned), then all the available channels become candidates. If only one of the end nodes of the extracted link has available radios, then we may select a channel from among those assigned to the radios of the other end node. If both end nodes have no available radios, we can only select channels in common to the end nodes. The initialization of one radio of every node to channel 1 ensures that every pair of end nodes share at least one common channel, hence the set $\mathcal{S}$ is always non-empty. The *optimization* step (lines 8–9), which is described hereinafter, preserves this condition despite it can remove the initial assignment of channel 1 to the radios of the end nodes.

For each channel $c \in \mathcal{S}$ we consider all the links $x \xrightarrow{c} y$ which would have $u \xrightarrow{c} v$ in their collision domain and compute the total utilization of their collision domain. In case a link $u \xrightarrow{c} v$ were established, all such total utilizations would be increased of the same amount, i.e., $\frac{f(u\xrightarrow{c}v)}{c(u\xrightarrow{c}v)}$. To the purpose of determining the channel and the rate which minimize the maximum total utilization, we only consider the maximum among such total utilizations, which we denote by $U'_{\max}(c)$ (line 19). If a link $u \xrightarrow{c} v$ were established, we would also need to consider the total utilization of its collision domain. If $U_{\text{tot}}(u \xrightarrow{c} v)$ (computed without the term $\frac{f(u\xrightarrow{c}v)}{c(u\xrightarrow{c}v)}$) is greater than $U'_{\max}(c)$, then we try to decrease the transmission rate on $u \xrightarrow{c} v$ in the attempt to reduce the size of the collision domain of $u \xrightarrow{c} v$ and hence its total utilization. We keep on trying lower transmission rates as long as $U_{\text{tot}}(u \xrightarrow{c} v)$ remains greater than $U'_{\max}(c)$. Also, the transmission rate is actually decreased if it allows to reduce

FCRA($G_I(V, E_I), \{f(e)\}_{e\in E_I}$)

1 $\mathcal{A}(u) \leftarrow \{1\} \quad \forall u \in V$

2 $E \leftarrow \left\{u \xrightarrow{1} v \,|\, u \rightarrow v \in E_I\right\}$

3 $c(e) \leftarrow r_M,\ U_{tot}(e) \leftarrow \sum_{e_0\in\mathcal{D}_{coll}(e)} \frac{f(e_0)}{c(e_0)} \quad \forall e \in E$

4 $Q \leftarrow \{e\}_{e\in E_I}$

5 **while** Q is not empty

6 **do** $(u \rightarrow v) \leftarrow$ EXTRACT_MAX(Q)

7 $E \leftarrow E - \left\{u \xrightarrow{c_T} v\right\}$

8 OPTIMIZE($G_I(V, E_I), u, Q$)

9 OPTIMIZE($G_I(V, E_I), v, Q$)

10 $\mathcal{S}_C \leftarrow \mathcal{A}(u) \cap \mathcal{A}(v)$

11 **if** $|\mathcal{A}(u)| < k(u)$ and $|\mathcal{A}(v)| < k(v)$

12 **then** $\mathcal{S} \leftarrow \mathcal{C}$

13 **elseif** $|\mathcal{A}(u)| = k(u)$ and $|\mathcal{A}(v)| < k(v)$

14 **then** $\mathcal{S} \leftarrow \mathcal{A}(u)$

15 **elseif** $|\mathcal{A}(u)| < k(u)$ and $|\mathcal{A}(v)| = k(v)$

16 **then** $\mathcal{S} \leftarrow \mathcal{A}(v)$

17 **else** $\mathcal{S} \leftarrow \mathcal{A}(u) \cap \mathcal{A}(v)$

18 **for** each $c \in \mathcal{S}$

19 **do** $U'_{max}(c) \leftarrow \max_{x \xrightarrow{c} y \,|\, u \rightarrow v \in \mathcal{N}(x \rightarrow y)} U_{tot}(x \xrightarrow{c} y)$

20 $m \leftarrow M,\ c(u \xrightarrow{c} v) \leftarrow r_M$

21 $m_{min} \leftarrow M,\ min \leftarrow U_{tot}(u \xrightarrow{c} v)$

22 **while** $m > 1$ and $U_{tot}(u \xrightarrow{c} v) > U'_{max}(c)$

23 **do** $m \leftarrow m - 1,\ c(u \xrightarrow{c} v) \leftarrow r_m$

24 **if** $U_{tot}(u \xrightarrow{c} v) < min$

25 **then** $min \leftarrow U_{tot}(u \xrightarrow{c} v)$

26 $m_{min} \leftarrow m$

27 $c(u \xrightarrow{c} v) \leftarrow r_{m_{min}}$

28 $U_{max}(c) \leftarrow \max\left(U'_{max}(c),\ U_{tot}(u \xrightarrow{c} v)\right)$

29 $\{c_i\}_{i=0}^{|\mathcal{S}_C|} \leftarrow \mathcal{S}_C \cup \operatorname{argmin}_{c\in\mathcal{S}-\mathcal{S}_C} U_{max}(c)$

▷ Sort $\{c_i\}$ in increasing order of $U_{max}(c_i)$

30 $F \leftarrow f(u \rightarrow v),\ i \leftarrow 0$

31 **while** $F > 0$

32 **do** $E \leftarrow E \cup \{u \xrightarrow{c_i} v\}$

33 $\mathcal{A}(u) \leftarrow \mathcal{A}(u) \cup \{c_i\},\ \mathcal{A}(v) \leftarrow \mathcal{A}(v) \cup \{c_i\}$

34 **if** $i < |\mathcal{S}_C|$ and $F \geqslant [U_{max}(c_{i+1}) - U_{max}(c_i)] \cdot \sum_{k=0}^{i} c(u \xrightarrow{c_k} v)$

35 **then** $I \leftarrow U_{max}(c_{i+1}) - U_{max}(c_i)$

36 **else** $I \leftarrow \frac{F}{\sum_{k=0}^{i} c(u \xrightarrow{c_k} v)}$

37 **for** $k \leftarrow 0$ to i

38 **do** $f(u \xrightarrow{c_k} v)$ += $I \cdot c(u \xrightarrow{c_k} v)$

39 F -= $I \cdot c(u \xrightarrow{c_k} v)$

40 $i \leftarrow i + 1$

▷ Update $U_{tot}(e) \quad \forall e \in E$

Fig. 2. Pseudo-code FCRA.

the total utilization (lines 20–27). Then, $U_{\max}(c)$ is set to the maximum between $U'_{\max}(c)$ and $U_{\text{tot}}(u \xrightarrow{c} v)$.

The pre-computed flow rate $f(u \rightarrow v)$ can then be split over links established on the candidate channels. Since we allow at most one radio on each end node to be assigned a new channel, we can only consider one channel that is not already shared by the end nodes. We select the channel that minimizes $U_{\max}(c)$ for $c \in \mathcal{S} - \mathcal{S}_C$. Such a channel and the channels in S_C constitute the actual set $\{c_i\}$ of candidate channels. In order to minimize the maximum total utilization, the flow rate $f(u \rightarrow v)$ is split in the following way. We begin allocating the flow rate on the candidate channel associated with the smallest $U_{\max}$ value (c_0, if we assume the set $\{c_i\}$ ordered for increasing values of $U_{\max}$) until either $U_{\text{tot}}(u \xrightarrow{c_0} v)$ equals $U_{\max}(c_1)$ or all the flow rate has been allocated. In the latter case, only the link $u \xrightarrow{c_0} v$ is established

OPTIMIZE$(G_I(V, E_I), u, Q)$
1 **if** $1 \notin \mathcal{A}(u)$ or $|\mathcal{A}(u)| < k(u)$
2   **then return**
3 $\mathcal{L} \leftarrow \{x \rightarrow y \in E_I \,|\, x = u \vee y = u\}$
4 **for** each $x \rightarrow y \in L - Q$
5   **do if** $x \xrightarrow{1} y \in E$
6   **then return**
7 **for** each $x \rightarrow y \in L \cap Q$
8   **do if** $\mathcal{A}(x) \cap \mathcal{A}(y) - \{1\} = \varnothing$
9   **then return**
10 **for** each $x \rightarrow y \in L \cap Q$
11   **do** select $c \in \mathcal{A}(x) \cap \mathcal{A}(y) - \{1\}$
12         $E \leftarrow E \cup \{x \xrightarrow{c} y\} - \{x \xrightarrow{1} y\}$
13 $\mathcal{A}(u) \leftarrow \mathcal{A}(u) - \{1\}$
    $\triangleright$ Update $U_{tot}(e) \quad \forall e \in E$

Fig. 3. Pseudo-code of the optimization step.

between u and v. In the former case, we keep on allocating the flow rate equally on c_0 and c_1 until either $U_{\text{tot}}(u \xrightarrow{c_0} v)$ and $U_{\text{tot}}(u \xrightarrow{c_1} v)$ equal $U_{\max}(c_2)$ or all the flow rate has been allocated, and so on. At the end of these steps (lines 30–40), the links $u \xrightarrow{c_i} v$ on which part of the pre-computed flow rate $f(u \rightarrow v)$ has been allocated are included in the set E of links of the graph G.

After $|E_I|$ iterations of the **While** loop, every link in E_I is replaced by at least one link in E.

1) Optimization Step: In the initialization phase of the FCRA algorithm one radio of every node is assigned channel 1. Given the limited availability of radios, this choice may be too wasteful. The optimization step is intended to check whether appropriate conditions apply to the end nodes of the extracted link that enable to remove the initial assignment of channel 1 while still ensuring that every pair of connected nodes share at least a common channel.

The OPTIMIZE$(G_I(V, E_I), u, Q)$ function (Fig. 3) is passed one end node of the extracted link (u) and the priority queue, and checks whether it is possible to remove channel 1 from u's radios. If none of u's radios is assigned channel 1 (node u might have already been passed to Optimize as end node of a previously extracted link) the function returns. OPTIMIZE also returns in case u has radios with no channel assigned, since removing channel 1 from a radio would not enlarge the set of candidate channels.

To remove channel 1 from u's radios while preserving the property that every pair of connected nodes in the network share a common channel, we need to analyze the set $\mathcal{L}$ of the links leaving or entering node u. We distinguish two cases:

- the link has already been extracted from Q (lines 4–6): such a link must not have been assigned channel 1, otherwise it is not possible to remove channel 1 from u's radios;
- the link has not been extracted from Q yet (lines 7–9): the end nodes must share at least one channel other than channel 1, otherwise it is not possible to remove channel 1 from u's radios.

The OPTIMIZE function returns as soon as a link is found which does not obey the above conditions. If no such link is found, channel 1 can be removed from among u's radios. In such case, we need to assign a new temporary channel to all the links leaving or entering u that are still in the queue Q.

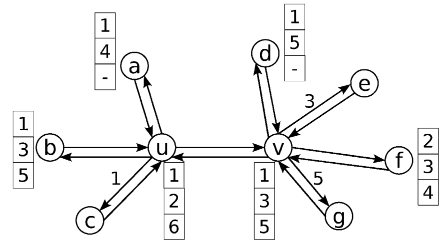

Fig. 4. Example to illustrate the optimization step.

Fig. 4 shows an example where link $u \rightarrow v$ has been just extracted. Numbers reported next to a node represent channels assigned to its radios, while the number close to a link represents the channel it has been assigned (only for already extracted links). As we can easily verify, channel 1 can be removed from v's radios but not from u's radios.

2) FCRA Complexity: The initialization phase requires $O(|V| + |E_I|^2)$ time, as computing the total utilization of the collision domain of a link requires to visit all the links in its potential collision domain. The **While** loop is repeated exactly $|E_I|$ times and the complexity of each iteration is $O(|V||E_I|)$. Indeed, the complexity of an operation on the priority queue can be made logarithmic (e.g., by using a heap) and the time required to compute $\mathcal{S}$ (lines 11–17) is $O(1)$ if we consider the number of channels as a constant. Computing $U'_{\max}(c)$ requires $O(|E_I|)$ if we store the total utilization values for all the collision domains and keep them up to date. Computing $U_{\max}(c)$ requires $O(|E_I|)$, while the last **While** loop counts for $O(1)$. Since new links have been added to E, we need to update the total utilization of the collision domains including the new links by adding the ratio of the flow to the rate of such links. Updating all the total utilization values takes $O(|E_I|)$ time. The OPTIMIZE function replaces channel 1 with a new temporary channel for every incident link that has not been extracted yet. Updating the total utilization values of all the collision domains affected by switching channel on a link takes $O(|E_I|)$ time, hence the complexity of the optimization step is $O(|V||E_I|)$. We conclude that the complexity of the FCRA algorithm is $O(|V||E_I|^2)$. If we determine the pre-computed flow rates as described in the previous section, we need to consider an additional $O(|V||E_I| \log \frac{|V|^2}{|E_I|})$ time if we compute the maxflow using the algorithm in [19].

C. Adjust the Pre-Computed Flow Rates

We have already shown in Section III that, in case the returned channel assignment is such that the sufficient condition (2) for schedulability is not satisfied, a set of schedulable flow rates can be obtained by dividing all the pre-computed flow rates by the maximum total utilization. However, if we re-run the channel assignment algorithm with the set of schedulable flow rates as input and obtain a different channel assignment, it means that the previously computed channel assignment does not best match the set of schedulable flow rates. Such an event cannot occur with the FCRA algorithm, as it holds the following property:

If $\mathcal{A}$ is the channel assignment and $\{U_{\text{tot}}(e)\}_{e \in E_I}$ is the set of values representing the total utilization of all the collision domains that the FCRA algorithm returns when it is passed *the set $\{f(e)\}_{e\in E_I}$ of pre-computed flow rates, then the same channel assignment $\mathcal{A}$ and the set $\{\alpha U_{\text{tot}}(e)\}_{e\in E_I}$ of total utilizations are returned when the FCRA algorithm is passed the set $\{\alpha f(e)\}_{e\in E_I}$ of pre-computed flow rates.*

This property comes from the fact that only comparisons among sums of pre-computed flow rates are done by the FCRA algorithm. As a consequence, if we scale the pre-computed flow rates to make them schedulable, we are assured that the returned channel assignment is the one that best fits the new set of schedulable flow rates.

V. A Layer-2.5 Forwarding Paradigm

A solution to the joint channel assignment and routing problem provides a set of link flow rates that are schedulable given the computed channel assignment. We develop a novel Layer-2.5 (L2.5) forwarding paradigm which enables every router to utilize each of its links in proportion to their flow rates. The operations of our forwarding paradigm do not involve the use of destination-based routing tables. In traditional distance-vector and link-state routing protocols, links are assigned a cost and some information on the network topology is disseminated among routers. Each router then builds its own routing table and forwards a packet along the minimum cost path to the destination of the packet. Thus, the forwarding decisions taken by a router depend on the cost of all the network links. This implies that, given the traffic demands, link costs need to be carefully computed to enable every router to approximate the flow rates on all of its link. However, a variation of the traffic demands may require to update the link costs, causing a considerable overhead. Also, in the event of temporary node/link failures, it is necessary to inform all the routers and wait for them to re-compute the routing tables.

According to our Layer-2.5 forwarding paradigm, the cost a router assigns to its links does not affect the forwarding decisions of other routers. Thus, each router can dynamically update its link costs to cope with changing traffic demands, without having to distribute them among routers. Also, in case a link quality degrades so as to make a neighbor unreachable, a router can exclude that neighbor and send packets to the other neighbors as soon as it perceives the new conditions.

The operations of our Layer-2.5 forwarding paradigm are described hereinafter. Each router is configured with the (static) set of schedulable flow rates associated with its links. Such values are used to take forwarding decisions as follows. Each router u sets a timer which expires at regular intervals and records the amount of bytes $b(v)$ sent to each neighbor v since the last timer expiration. Every time u has to take a forwarding decision, it computes the following value for each neighbor v:

$$\Delta_u(v) = \frac{f(u\to v)}{\sum_{\forall u\to i} f(u\to i)} - \frac{b(v)}{\sum_{\forall u\to i} b(i)}$$

$\Delta_u(v)$ represents the gap between the desired and the current utilization of link $u\to v$ and can be seen as the cost of that link. In the attempt to keep the transmission rate on all of its links proportional to the corresponding flow rate, router u will send the packet to the neighbor node having the largest $\Delta_u(v)$ value, i.e., the maximum cost.

Clearly, a router cannot decide the next hop of a packet based only on such link costs, as packets could take extremely long paths to get to destination. Routers must also take their decisions in such a way that packets advance towards their destination. It is therefore necessary to make routers aware of how "far" they are from the destination of the packet. For this purpose, we do not require each router to have a complete knowledge of the network topology. The information available to each router simply consists in the minimum hop count to each destination. Possible destinations are all the aggregation devices and the supersink representing any gateway. We denote by $\vec{\text{HC}}_u$ the *hop count* vector of node u, i.e., the set of the minimum hop counts from u to all the destinations.

Routers build their own hop count vector in much the same way as described in [20]. Every aggregation device and every gateway initially broadcast (through all of their radios) an advertisement message containing the variable $\text{HC}_d(d) = 0$ representing the hop count to node d, where d is the unique identifier of the router sending the advertisement message. In particular, gateways do not use their own identifier, but a value that is used to refer to the supersink. Each node u collects all incoming messages, sets its own $\text{HC}_u(d)$ variables to the smallest received values plus 1 and broadcasts a new message advertising $\text{HC}_u(d)$. The procedure ends when all the routers have received and forwarded messages for all the destinations. We also adopt the strategy based on timers proposed in [20] to limit the flooding of advertisement messages. Unlike [20], however, we do not use the hop count vector to forward packets along the shortest path.

While building its own hop count vector, each router also stores the messages advertised by its neighbors. This information is used to partition the neighbors based on the hop count to a certain destination. For a generic destination d, the set $\mathfrak{N}(u)$ of neighbors of node u is partitioned into three sets: $\mathfrak{N}_d^{=}(u)$ (containing the neighbors with the same hop count to d as u), $\mathfrak{N}_d^{+}(u)$ (containing the neighbors with u's hop count to d plus 1) and $\mathfrak{N}_d^{-}(u)$ (containing the neighbors with u's hop count to $d-1$).

The proposed Layer-2.5 forwarding paradigm provides that every source node s (aggregation devices and gateways) includes a *maximum hop count* $\text{HC}_s^{\max}$ in each packet it sends. Such value represents the maximum allowable number of hops to the destination and can be the minimum hop count to the destination multiplied by a constant factor $\alpha > 1$. This value is decremented at each intermediate hop and is used to determine the set $\mathfrak{S}$ of candidate next hop neighbors. Indeed, if the maximum hop count equals the minimum hop count to the destination for the intermediate node u, then the packet must be necessarily sent to a neighbor in $\mathfrak{N}_d^{-}(u)$ and will thus follow a minimum hop path. Otherwise, the packet may also be sent to neighbors in $\mathfrak{N}_d^{=}(u)$ or even $\mathfrak{N}_d^{+}(u)$. The neighbor which the packet is received from is excluded from $\mathfrak{S}$. More precisely, a router u receiving a packet from node w with maximum hop count $\text{HC}_u^{\max}$ and destined to node d, determines the set $\mathfrak{S}$ of candidate next hop neighbors as follows:

$$\mathfrak{S} = \begin{cases} \mathfrak{N}(u) - \{w\} & \text{if } \text{HC}_u^{\max} > \text{HC}_u(d)+1, \\ \mathfrak{N}_d^{=}(u) \cup \mathfrak{N}_d^{-}(u) - \{w\} & \text{if } \text{HC}_u^{\max} = \text{HC}_u(d)+1, \\ \mathfrak{N}_d^{-}(u) - \{w\} & \text{if } \text{HC}_u^{\max} = \text{HC}_u(d) \end{cases}. \tag{3}$$

Router u then computes $\Delta_u(v)$ for all neighbors $v \in \mathfrak{S}$ as described earlier, decrements $\mathrm{HC}^{\max}$ and sends the packet to the neighbor $v \in \mathfrak{S}$ with the maximum $\Delta_u(v)$.

This strategy ensures that packets reach the destination in at most maximum hop count hops. However, since a router u can select a neighbor in $\mathfrak{N}_d^+(u)$ or $\mathfrak{N}_d^=(u)$ as long as $\mathrm{HC}_u^{\max}$ exceeds the minimum hop count to the destination, it is likely that packets reach the destination in exactly maximum hop count hops. This behavior is confirmed by the simulation study presented next. To reduce the average path length and hence the resource consumption, we introduce three functions to weigh the flow rates of the links of a router. The rationale is to apply a weight (≤ 1) to the flow rates of the links to neighbors in $\mathfrak{N}_d^+(u)$ and $\mathfrak{N}_d^=(u)$, so as to decrease the corresponding $\Delta_u(v)$ and hence the probability for such neighbors to be selected. The weight is lower as $\mathrm{HC}_u^{\max}$ approaches the minimum hop count to the destination. More precisely, the link cost is computed as

$$\Delta_u(v) = \frac{\beta_{uv}\left(\frac{\mathrm{HC}_u^{\max}}{\mathrm{HC}_u(d)+1}\right) \cdot f(u \to v)}{\sum_{\forall u \to i} \beta_{ui}\left(\frac{\mathrm{HC}_u^{\max}}{\mathrm{HC}_u(d)+1}\right) \cdot f(u \to i)} - \frac{b(v)}{\sum_{\forall u \to i} b(i)} \quad (4)$$

where

$$\beta_{uv}(x) = \begin{cases} \left(\frac{x-1}{x}\right)^{2\beta} \cdot \mathbf{1}_{[1,+\infty]}(x) & \text{if } v \in \mathfrak{N}_d^+(u), \\ \left(\frac{x-1}{x}\right)^{\beta} \cdot \mathbf{1}_{[1,+\infty]}(x) & \text{if } v \in \mathfrak{N}_d^=(u), \\ 1 & \text{if } v \in \mathfrak{N}_d^-(u) \end{cases}$$

and $\mathbf{1}_{[1,+\infty]}(x)$ is an indicator function that yields 1 if $x \geq 1$ and 0 otherwise. The β parameter impacts the weight applied to the flow rates. The higher β, the lower the weight. We note that the flow rates of the links to neighbors in $\mathfrak{N}_d^+(u)$ are decreased more than those of neighbors in $\mathfrak{N}_d^=(u)$. Also, if $\beta = 0$ all the weights are unitary.

Each router selects the next hop neighbor on a per-packet basis, with no pre-computed routes available. Hence, our forwarding paradigm has the potential for a fast recovery from link failures. For instance, if the neighbor selected as the next hop for a packet becomes unreachable and several repeated transmissions fail (no acknowledgment is received), then such neighbor is removed from the candidate set $\mathfrak{S}$ and a new next hop neighbor is selected (without the need to update the hop count vectors). However, selecting the next hop neighbor on a per-packet basis has also the undesired effect to split flows along multiple paths, thus requiring the reordering of packets at destination. Countermeasures to such effect are under investigation and may consist in adding flow identification tags in the packets. Future work also includes the opportunity to further weigh the flow rates to take the quality of a link into account in the selection of the next hop neighbor.

A. Markov Chain Model of the L2.5 Forwarding Paradigm

To gain more insight into the impact of parameters α and β on the behavior of the proposed Layer-2.5 forwarding paradigm, we model its operation by means of a bi-dimensional Markov chain. To this end, we need to appropriately define the state variables and compute the transition probabilities. We introduce the bi-dimensional state variable $x_n = (i; h)$ indicating that the packet is at node i at time n with a maximum hop count $\mathrm{HC}_i^{\max} = h$. The transition probability $\Pr\{x_{n+1} = (i;h)|x_n=(j;k)\}$ is clearly null for $h \neq k-1$, as the maximum hop count is decremented at each hop. Otherwise, the transition probability depends on the value of k, which determines the set $\mathfrak{S}$ of candidate next hop nodes (3). We recall that the selected next hop is the node $v \in \mathfrak{S}$ with the maximum $\Delta_u(v)$ (4). The probability of selecting a node i given that the packet is at node j is approximated as

$$\begin{aligned}\Pr\{x_{n+1} = (i;h)|x_n = (j;h+1)\} \\ = \frac{\beta_{ji}\left(\frac{h+1}{\mathrm{HC}_j(d)+1}\right) \cdot f(j \to i)}{\sum_{\forall j \to u} \beta_{ju}\left(\frac{h+1}{\mathrm{HC}_j(d)+1}\right) f(j \to u)}\end{aligned} \quad (5)$$

We note that the weight β_{ju} is null if $u \notin \mathfrak{S}$, thus the above probability correctly evaluates as 0 if $i \notin \mathfrak{S}$ and the sum of such probabilities over all the neighbors is 1. (5) holds for $j \neq d$. Once the packet arrives at the destination node d, it will be routed outside the mesh network. In our model, we assume the packet stays at node d, i.e., $\Pr\{x_{n+1} = (i;h)|x_n = (d;h+1)\} = 0 \quad \forall i \neq d$ and $\Pr\{x_{n+1} = (d;h)|x_n = (d;h+1)\} = 1$.

The transition probabilities are independent of the time n (they are said to be *time-homogeneous*). Hence, without ambiguity, we omit the indication of the time and use the notation $\Pr\{(i;h)|(j;h+1)\}$ to refer to a transition probability.

We point out that using the node which holds the packet at time n as the state variable (i.e., $x_n = i$) does not assure the Markov property. Indeed, the future state would depend not only on the present state but also on the past states, as the information about the maximum hop count could not be derived from the present state. This is the reason to consider a bi-dimensional state variable which also embeds the information about the maximum hop count. In this way, the future state is conditionally independent of the past states given the present state.

Given the transition probabilities, we can compute the state probabilities at time n by using the law of total probability:

$$\begin{aligned}\Pr\{x_n = (i;h)\} = \sum_{j \in V} \Pr\{x_n = (i;h)|x_{n-1} = (j;h+1)\} \\ \cdot \Pr\{x_{n-1} = (j;h+1)\}\end{aligned} \quad (6)$$

By defining the $|V| \times 1$ vector

$$\bar{p}_n(h) = \begin{bmatrix} \Pr\{x_n{=}(0;h)\} \\ \Pr\{x_n{=}(1;h)\} \\ \vdots \\ \Pr\{x_n{=}(|V|-1;h)\} \end{bmatrix}$$

and the $|V| \times |V|$ matrix $\bar{\bar{R}}$

$$\bar{\bar{R}}_{ij}(h) = \Pr\{(i;h)|(j;h+1)\}$$

we can write (6) in the following vector-matrix form:

$$\begin{aligned}\bar{p}_n(h) &= \bar{\bar{R}}(h) \cdot \bar{p}_{n-1}(h+1) \\ &= \bar{\bar{R}}(h) \cdot \bar{\bar{R}}(h+1) \cdot \bar{p}_{n-2}(h+2) \\ &= \prod_{m=0}^{n-1} \bar{\bar{R}}(h+m) \cdot \bar{p}_0(h+n)\end{aligned} \quad (7)$$

where $\bar{p}_0(h)$ represents the initial distribution. We assume that each source node generating a packet sets the maximum hop count to its minimum hop count to the destination times α. Hence for every source node $i \in V_A$, $\Pr\{x_0=(i;h)\} = 0$ if $h \neq \lfloor \alpha \mathrm{HC}_i(d) \rfloor$ and $\Pr\{x_0 = (i; \lfloor \alpha \mathrm{HC}_i(d) \rfloor)\} = q_0(i)$, where the $q_0(i)$ values are such that $\sum_{i \in V_A} q_0(i) = 1$.

We now evaluate the number of hops taken by a packet to reach the destination. We introduce the following random variable:

$$n_d = \min\{n : x_n = (d; \star)\}$$

where $(d;\star)$ represents any of the states with d as node. Possible outcomes of such a random variable are the integers between 0 and $\max_{i \in V_A} \lfloor \alpha \mathrm{HC}_i(d) \rfloor$. For any value m in such interval, we want to determine the probability that the packet reaches the destination in m hops. Using again the law of total probability yields

$$\begin{aligned}\Pr\{n_d = m\} &= \sum_h \sum_{j \in V} \Pr\{n_d = m \,|\, x_{m-1} = (j;h)\} \\ &\quad \cdot \Pr\{x_{m-1} = (j;h)\} \\ &= \sum_h \sum_{j \neq d} \Pr\{x_m = (d; h-1) \,|\, x_{m-1} = (j;h)\} \\ &\quad \cdot \Pr\{x_{m-1} = (j;h)\}\end{aligned}$$

The last equality holds because $\Pr\{n_d = m | x_{m-1} = (j;h)\}$ equals 0 if $j = d$ (the packet is already at the destination at time $m-1$) and the probability that the packet is forwarded to the destination if the packet has not yet reached the destination at time $m-1$. Since $\Pr\{(d;h-1)|(d;h)\} = 1$, (6) yields

$$\begin{aligned}\Pr\{x_m = (d;h-1)\} &= \Pr\{x_{m-1}=(d;h)\} \\ &+ \sum_{j \neq d} \Pr\{x_m = (d;h-1) \,|\, x_{m-1} = (j;h)\} \\ &\cdot \Pr\{x_{m-1} = (j;h)\}\end{aligned}$$

hence

$$\begin{aligned}\Pr\{n_d = m\} &= \sum_h \Pr\{x_m=(d;h-1)\} \\ &\quad - \sum_h \Pr\{x_{m-1} = (d;h)\}\end{aligned} \tag{8}$$

Intuitively, the probability that the packet takes exactly m hops to reach the destination is the difference between the probability that the packet is at the destination at time m and the probability that the packet is at the destination at time $m-1$. This result is used to compute the *average normalized path length* (ANPL) associated with a source s, i.e., the average length of the paths taken by packets sent by s normalized to the minimum hop count of s to the destination:

$$\mathrm{ANPL}(s) = \frac{\sum_{0 \le m \le \lfloor \alpha \mathrm{HC}_s(d) \rfloor} m \cdot \Pr\{n_d = m\}}{\mathrm{HC}_s(d)} \tag{9}$$

We now want to determine the probability that a node j sends a packet to node i independently from the maximum hop count included in the packet. In other words, we want to determine the probability $\Pr\{i \,|\, j\}$ that a packet leaves any of the states with j as node and enters any of the states with i as node. A packet arrived at j may have been originated at any of the source nodes. If s denotes the source node, the maximum hop count included in the packet can range from 0 to $\lfloor \alpha \mathrm{HC}_s(d) \rfloor$. Thus, the law of total probability yields

$$\Pr\{i \,|\, j\} = \sum_{\substack{s \in V_A \\ 0 \le h \le \lfloor \alpha \mathrm{HC}_s(d) \rfloor}} \Pr\{i \,|\, j, h, s\} \cdot \Pr\{h, s \,|\, j\} \tag{10}$$

Since the next state is conditionally independent of the node which originated the packet given the present state, $\Pr\{i|j,h,s\}$ equals the transition probability $\Pr\{(i;h-1)|\ (j;h)\}$. The Bayes' theorem also yields

$$\Pr\{h, s \,|\, j\} = \frac{\Pr\{j \,|\, h, s\} \cdot \Pr\{h, s\}}{\sum_{\substack{s^\star \in V_A \\ 0 \le h^\star \le \lfloor \alpha \mathrm{HC}_{s^\star}(d) \rfloor}} \Pr\{j \,|\, h^\star, s^\star\} \cdot \Pr\{h^\star, s^\star\}}$$

If a packet has been originated at node s and arrives at j with a maximum hop count of h, it means that the packet has taken $\lfloor \alpha \mathrm{HC}_s(d) - h \rfloor$ hops to reach j. Thus, $\Pr\{j \,|\, h, s\} = \Pr\{x_{\lfloor \alpha \mathrm{HC}_s(d) - h \rfloor} = (j;h) \,|\, s\}$. Finally,

$$\begin{aligned}\Pr\{h, s\} &= \Pr\{h \,|\, s\} \cdot \Pr\{s\} \\ &= \frac{q_0(s)}{\lfloor \alpha \mathrm{HC}_s(d) \rfloor + 1}\end{aligned}$$

assuming that the maximum hop count included in the packet takes all the values between 0 and $\lfloor \alpha \mathrm{HC}_s(d) \rfloor$ with the same probability, given that the source node is s.

We performed numerical simulations based on this Markov chain model to determine proper values for the parameters α and β. Also, ns-2 simulations allowed to verify the accuracy of this model. The results of these studies are presented next.

VI. Performance Evaluation

We performed a number of simulation studies to evaluate the performance of both the centralized channel and rate assignment algorithm and the Layer-2.5 forwarding paradigm. The results are illustrated in the following subsections.

A. FCRA: Performance Evaluation

As described in the Introduction, the channel assignment algorithms which are part of a heuristic for the joint channel assignment and routing problem address the problem to assign channels to radios in order to make a given set of pre-computed flow rates schedulable. In Section III, we have also shown that the channel assignment problem is NP-complete and the minimum scaling factor which yields a set of flow rates satisfying the sufficient condition for schedulability is given by the maximum among the total utilization of all the collision domains. The aim of this section is thus to evaluate the effectiveness of different channel assignment algorithms by comparing the resulting minimum scaling factor $\lambda = \max_{e \in E} U_{\mathrm{tot}}(e)$. We compare our FCRA algorithm to LACA (Load-Aware Channel Assignment) [5] and BSCA (Balanced Static Channel Assignment) [11]. We also consider two variants of FCRA, FCRA_noRA (no rate adaptation is performed) and FCRA_noOpt (the optimization step is not performed), to evaluate the performance increase due to adapting the transmission rate and performing the optimization step.

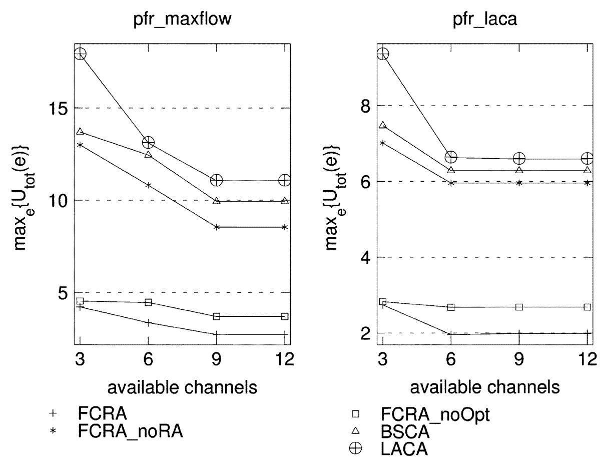

Fig. 5. 25-nodes topologies.

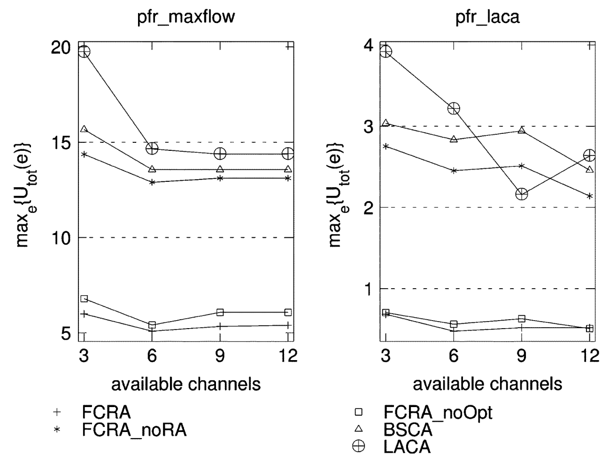

Fig. 6. 50-nodes topologies.

We consider two classes of topologies with 25 and 50 nodes, respectively, and a different distribution of radios per node. For the first (second) class, we generate 20 connected topologies by randomly placing nodes in a 300×300 (400×400) square meter area. The potential communication graph is then built using the following settings. The gain of the radio channel between two nodes is inversely proportional to the square of their distance. The thresholds γ_{r_m} are determined by using the *BER-SINR* curves of the IEEE 802.11a/g modulation schemes [15] with target bit-error-rate (BER) of 10^{-6} and packet size of 1024 bytes. The thermal noise is set to the same constant for all the nodes. The transmission power (the same for all the radios) is computed (using condition (1)) such that a link between two nodes exists if their distance is less than 90 m. The link capacities are set according to the values commonly advertised by 802.11a vendors: 54 Mb/s when the end nodes are within 30 m, 48 Mb/s when within 32 m, 36 Mb/s when within 37 m, 24 Mb/s when within 45 m, 18 Mb/s when within 60 m, 12 Mb/s when within 69 m, 9 Mb/s when within 77 m, and 6 Mb/s when within 90 m. We remark that the FCRA algorithm (and its FCRA_noOpt variant) instead determines the link capacities in order to minimize the maximum total utilization, as described in Section IV-B.

We consider two scenarios differing in the way the set of pre-computed flow rates are computed. We denote by *pfr_maxflow* the method we proposed in Section IV-A, and by *pfr_laca* the method proposed in [5] to compute the set of initial flow rates. We also consider different numbers of available channels: 3, 6, 9, and 12.

The results of the simulations are shown in Figs. 5 and 6 for the 25 and 50 nodes topologies, respectively. Each figure illustrates both the cases where *pfr_maxflow* and *pfr_laca* are used. For each value of the total number of available channels, the figures show the average values of $\lambda = \max_{e \in E} U_{\text{tot}}(e)$ over all the 20 topologies of each class for every channel assignment algorithm. We can observe that the best performance is achieved by FCRA, as in every scenario it exhibits the smallest scaling factor required to obtain a set of schedulable flow rates. Also, the optimization step enables a slight increase in the performance of FCRA, except in the case only 3 channels are available. We expected such a behavior because if 3 channels are available and nodes have 2 or 3 radios, then every channel is likely to be used for one radio of every node and the attempt to remove the initial channel 1 brings no benefit. By observing the performances of FCRA_noRA, BSCA and LACA, it looks clear that adapting the transmission rates brings instead a considerable benefit. The maximum total utilization returned by FCRA is from 2 to 4 times smaller than that of the other algorithms. Even without adapting the transmission rates, our algorithm (FCRA_noRA) performs better than the others. Finally, we note that in the *pfr_laca* scenario for the 50 nodes topologies, FCRA is the only algorithm (along with FCRA_noOpt) which succeeds in finding a channel assignment that makes the given set of pre-computed flow rates schedulable, as the resulting maximum total utilization is below 1.

B. Model-Based Analysis of the L2.5 Forwarding Paradigm

The Markov chain model of the L2.5 forwarding paradigm supplies formulas to compute the average normalized path length associated with a source node (9) and the probability that a node j selects neighbor i as next hop for a packet (10). Since the goal of the L2.5 forwading paradigm is to utilize links in proportion to their flow rates, we consider the mean square error between $\Pr\{i \mid j\}$ and the normalized flow rate on link $j \rightarrow i$ to evaluate its effectiveness:

$$\mathrm{MSE}_{mcm} = \frac{\sum_{\forall j \rightarrow i \in E} \left(\Pr\{i \mid j\} - \frac{f(j \rightarrow i)}{\sum_{\forall j \rightarrow u} f(j \rightarrow u)} \right)^2}{|E|}$$

We use the subscript mcm to denote that a quantity is evaluated through the Markov chain model. We performed numerical simulations for different values of the parameters α and β to evaluate their impact on ANPL_{mcm} and MSE_{mcm}. The results for a 25 node topology in case 6 channels are available and the pre-computed flow rates are determined with the method we propose in Section IV-A are shown in Figs. 7(a) and (b). Very similar results have been obtained for the other topologies, different number of available channels and the *pfr_laca* method. We first comment on Fig. 7(b). In case $\beta = 0$, ANPL_{mcm} is very close to α, meaning that the average path length is very close to the maximum path length, which is α times the minimum hop count to the destination of the source

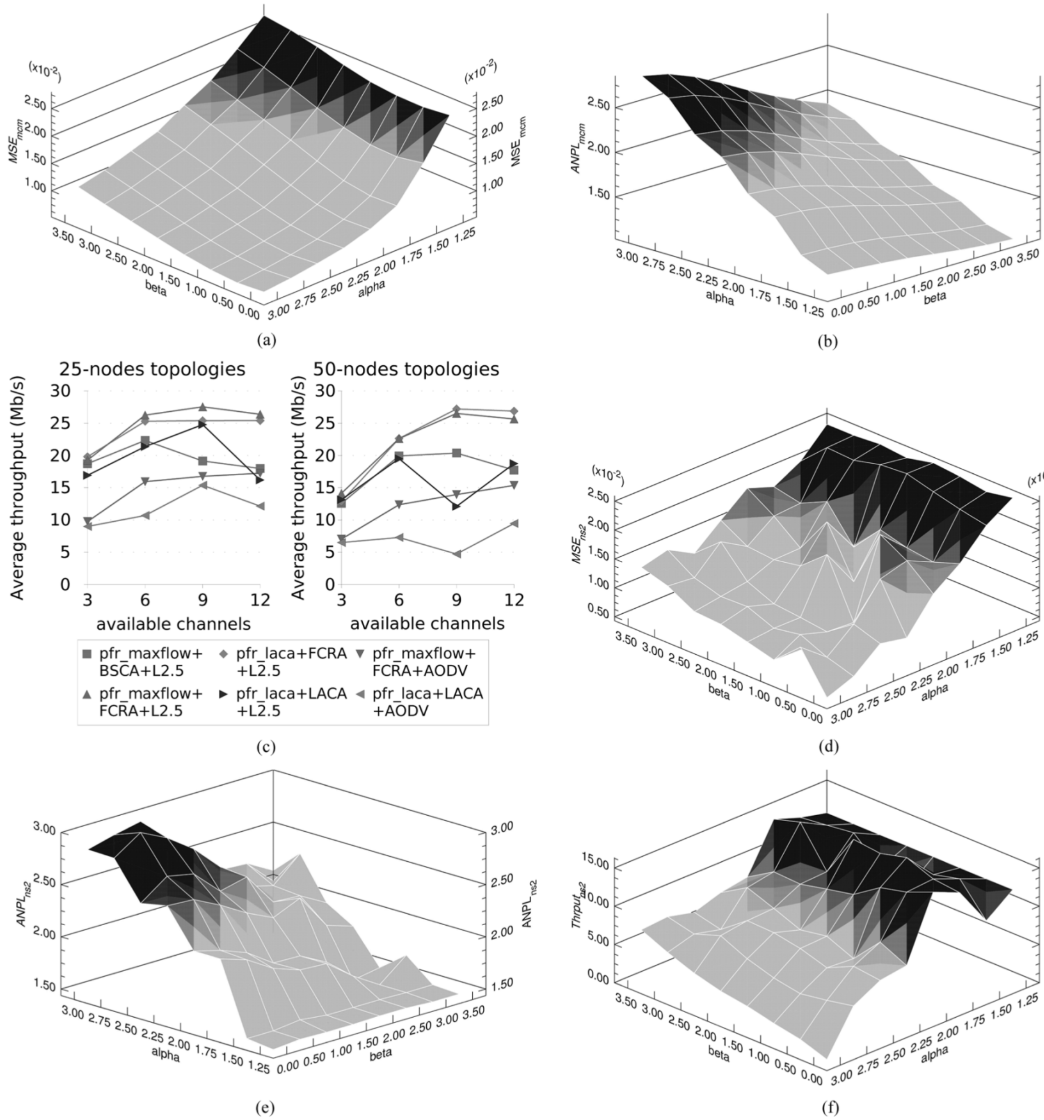

Fig. 7. Analysis and evaluation of the L2.5 forwarding paradigm (a) MSE resulting from the Markov chain model (b) ANPL resulting from the Markov chain model (c) Performance comparison (d) ANPL resulting from ns2 simulations (e) ANPL resulting from ns2 simulations (f) Throughput for different α and β values.

node. Increasing β instead causes ANPL_{mcm} to decrease, i.e., the average path length becomes smaller and smaller than the maximum path length. This behavior confirms our intuition that applying a proper weight to the link flow rates enables to reduce the average path length. The benefit of this strategy in terms of resource usage and network throughput is investigated in the next subsection. Fig. 7(a) shows that MSE_{mcm} decreases by increasing α and decreasing β. This behavior may be intuitively explained by considering that with higher values of α and lower values of β it is more likely that a packet includes a maximum hop count which is much larger than the minimum hop count to the destination of the intermediate node. In this case, all the neighbors are candidate next hops and all the weights are close to 1, hence the intermediate node can select the next hop in order to better match the flow rates on its links. Unfortunately, we have shown that high values of α and low values of β lead to the average path length being very close to the maximum path length. Besides showing the trade-off between having the link utilizations proportional to the flow rates and limiting the average path length, the numerical simulations based on the Markov chain model also provide a means to solve it. Indeed, we can constrain ANPL_{mcm} not to exceed a fixed value and choose the α and β values that minimize MSE_{mcm} among those such that ANPL_{mcm} is less than the fixed value. For instance, the minimum (unconstrained) MSE_{mcm} is 0.0052096 and is achieved for $\alpha = 3$ and $\beta = 0.5$. If we impose $\mathrm{ANPL}_{mcm} < 2$, the minimum (constrained) MSE_{mcm} is 0.0088021 and is achieved for $\alpha = 2$ and $\beta = 0.5$.

C. Ns2-Based Analysis of the L2.5 Forwarding Paradigm

This subsection analyzes the performance of our L2.5 forwarding paradigm for different values of α and β by means of simulations with the ns2 network simulator. We consider the same scenario as in the previous subsection and perform several tests where the aggregation devices generate exponential on-off TCP traffic destined to the supersink with different parameters. We measure the resulting average normalized path length and the MSE between the link utilization and the normalized flow rate. The utilization of a router's link is measured as the ratio of the number of packets forwarded on the link to the total number of packets forwarded by the router. The average results over all the tests are shown in Figs. 7(d) and (e). We observe that the MSE and ANPL measured through ns2 simulations show the behavior anticipated by the Markov chain model (Figs. 7(a) and 7(b)). We also performed ns2 simulations in other scenarios (different topologies, available channels and methods to compute the flow rates), obtaining the same results. This ns2 study thus confirms that the Markov chain model is able to predict the value of measures such as the MSE and the ANPL with a good approximation.

We also measure the network throughput for different traffic loads and present the average values (in Mb/s) in Fig. 7(f). As expected, for high values of α the throughput is low despite the link utilizations are close to the flow rates (low MSE), because packets take long paths and hence consume resources. Increasing β slightly helps, as the average path length decreases. Decreasing α brings a smaller maximum path length and a higher throughput. However, the maximum throughput is not achieved for the minimum α and the maximum β (i.e., when the average path length is minimum), but for $\alpha = 1.75$ and $\beta = 1$, meaning that enforcing the link flow rates is also important. For the values ($\alpha = 2$ and $\beta = 0.5$) derived from the model by taking the values that minimize MSE_{mcm} subject to the constraint $\mathrm{ANPL}_{mcm} < 2.$, the throughput is quite close to the maximum one (5% lower). The Markov chain model thus provides an effective technique for selecting values for the parameters of our L2.5 forwarding paradigm. We applied such a technique to determine the values used in the simulations presented next involving the L2.5 forwarding paradigm.

D. Performance Evaluation

In the previous subsections, we first studied the performance of FCRA and other channel assignment algorithms in terms of the minimum scaling factor to apply to a given set of pre-computed flow rates to yield a set of schedulable flow rates. Then, we studied how to tune the parameters of the L2.5 forwarding paradigm to have the link utilizations proportional to the schedulable flow rates, while limiting the average path length. In those studies, we used two different methods to determine the set of pre-computed flow rates (*pfr_maxflow* and *pfr_laca*). In this subsection we evaluate how different combinations of these methods and channel assignment algorithms perform in terms of network throughput. Indeed, different channel assignments lead to a different composition of the collision domains. To evaluate the performance of the L2.5 forwarding paradigm, we also conduct simulations with the AODV (Ad-hoc On-demand Distance Vector) [21] routing protocol.

For each ns2 simulation, traffic sources generate exponential on-off TCP traffic and the average throughput over the whole simulation is measured. Such value is then averaged over all the 20 topologies of each class of topologies (25- and 50-nodes). The results are shown in Fig. 7(c). We observe that, being the computed channel assignment equal ($\mathrm{pfr\_maxflow} + \mathrm{FCRA}$ or $\mathrm{pfr\_laca} + \mathrm{LACA}$), our L2.5 forwarding paradigm achieves a higher throughput than AODV. Thus, our forwarding paradigm, which utilizes links in proportion to schedulable flow rates, gives better results than a protocol which is unaware of the channel assignment. Also, being the forwarding/routing protocol equal, our proposals $(\mathrm{pfr\_maxflow} + \mathrm{FCRA})$ allow a higher throughput than $\mathrm{pfr\_laca} + \mathrm{LACA}$ [5]. Finally, we note (and this is confirmed by looking at the other combinations not shown here) that, being the method to determine the pre-computed flow rates and the forwarding/routing protocol equal, FCRA allows to achieve a higher throughput than LACA and BSCA.

VII. Conclusion

This paper addressed fundamental design issues in multi-radio wireless mesh networks. We proposed a method to determine the pre-computed flow rates based on the maxflow algorithm, in order to maximize the network throughput. We showed that not only the channels but also the transmission rates of the links have to be properly selected to make a given set of pre-computed flow rates schedulable. Thus, we proposed a greedy heuristic for the channel and rate assignment problem. We developed a novel Layer-2.5 forwarding paradigm, which enables each mesh router to autonomously take forwarding decisions such that the transmission rate on each link approximates a given flow rate. Also, we presented a Markov chain model of our forwarding paradigm that helps determine proper values for its parameters. We performed a thorough simulation study which showed that the FCRA algorithm has the best performance in minimizing the maximum total utilization, the L2.5 forwading paradigm succeeds in achieving the flow rates on the network links, and the combination of our proposals outperforms other schemes.

As for the future work, the FCRA algorithm may be extended to consider variable transmission powers. Also, some links may not be assigned a channel in order to further reduce the maximum total utilization. As far as the L2.5 forwarding paradigm, weighting factors to take link quality into account while selecting the next hop neighbor must be devised.

Appendix

Let $G(V, E)$ be an undirected graph and $\{f(e)\}_{e \in E_I}$ a set of pre-computed flow rates. For simplicity, we assume $c(e_0) = C \quad \forall e_0 \in E$. Hence, we ask whether a channel assignment exists such that $\max_{e \in E} \sum_{e_0 \in \mathcal{D}_{\mathrm{coll}}(e)} f(e_0) \leq C$. We prove that such decision problem is NP-complete. First, such a problem $\in$ NP since, given a channel assignment, it can be verified in polynomial time if it is a solution for our problem. Next, we reduce the *Multiple Subset Sum Problem* (MSSP) [22] to the channel assignment problem to prove that it is NP-hard. Given a set of n items with weights $W_1, W_2, \ldots W_n$, and a set of m identical bins of capacity C each, the MSSP is the problem to select a subset of items of maximum total weight that can be packed in the bins. The decision version of MSSP is the problem to find out if all items can be packed in the bins. We first show that any instance of MSSP can be converted in polynomial time to an instance of the channel assignment problem. We construct a single potential collision domain network of $2 * n$ nodes where each node is equipped with 2 radios. Then we add a link between nodes 1 and 2 with a pre-computed flow rate of W_1, a link between 3 and 4 with a pre-computed flow rate of W_2, and so on. We also add a link between nodes 2 and 3, nodes 4 and

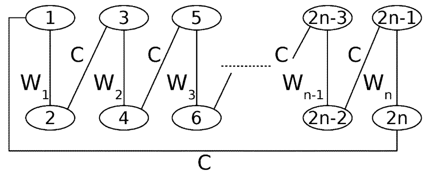

Fig. 8. Network graph constructed to prove that MSSP is reducible to the channel assignment problem.

5, and so on, and nodes $2n$ and 1, each with a pre-computed flow rate of $\mathcal{C}$ (Fig. 8). The capacity of each link is the same as the bin capacity C and the number of channels is equal to $m+n$. Now, we show that the instance of MSSP (decision version) is satisfiable if and only if the instance we constructed of the channel assignment problem is satisfiable. We note all the links with a pre-computed flow rate of C should not include any other link in their collision domain in order to satisfy the constraint $\sum_{e_0 \in \mathcal{D}_{\rm coll}(e)} f(e_0) \leq C$. Thus, they each must be assigned a different channel than all the other links. Hence, the remaining m channels must be assigned to the remaining n links with pre-computed flow rates equal to $W_1, W_2, \ldots W_n$. Then, a channel assignment such that $\max_{e \in E} \sum_{e_0 \in \mathcal{D}_{\rm coll}(e)} f(e_0) \leq C$ exists if and only if it is possible to pack all the n items in the m bins with capacity C. This concludes our proof.

REFERENCES

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

[2] Y. Liu and E. Knightly, "Opportunistic fair scheduling over multiple wireless channels," in *Proc. IEEE INFOCOM*, 2003, vol. 2, pp. 1106–1115.

[3] J. So and N. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proc. ACM MobiHoc*, 2004, pp. 222–233.

[4] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in *Proc. IEEE WCNC*, 2005, vol. 4, pp. 2051–2056.

[5] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM Mobile Comput. Commun. Rev.*, vol. 8, no. 2, pp. 50–65, Apr. 2004.

[6] A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proc. IEEE INFOCOM*, 2005, vol. 3, pp. 2223–2234.

[7] M. Alicherry, R. Bhatia, and E. Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 1960–1971, Nov. 2006.

[8] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and qos routing in multi-channel wireless mesh networks," in *Proc. ACM MobiHoc*, 2005, pp. 68–77.

[9] M. K. Marina and S. R. Das, "A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks," in *Proc. IEEE BroadNets*, 2005, vol. 1, pp. 381–390.

[10] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proc. IEEE INFOCOM*, 2006, pp. 1–12.

[11] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proc. ACM MobiCom*, 2005, pp. 73–87.

[12] A. H. Mohsenian Rad and V. W. S. Wong, "Joint optimal channel assignment and congestion control for multi-channel wireless mesh networks," in *Proc. IEEE ICC*, 2006, vol. 5, pp. 1984–1989.

[13] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 104–116, 2005.

[14] S. H. Low, "A duality model of tcp and queue management algorithms," *IEEE/ACM Trans. Networking*, vol. 11, no. 4, pp. 525–536, Aug. 2003.

[15] D. Qiao, S. Choi, and K. G. Shin, "Goodput analysis and link adaptation for ieee 802.11a wireless LANs," *IEEE Trans. Mobile Comput.*, vol. 1, no. 4, pp. 278–292, 2002.

[16] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[17] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proc. ACM MobiCom*, 2003, pp. 66–80.

[18] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications.*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[19] B. V. Cherkassky and A. V. Goldberg, "On implementing push-relabel method for the maximum flow problem," *Algorithmica*, vol. 19, pp. 390–410, 1997.

[20] F. Ye, A. Chen, S. Lu, and L. Zhang, "A scalable solution to minimum cost forwarding in large sensor networks," in *Proc. IEEE ICCCN*, 2001, pp. 304–309.

[21] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," IETF, RFC 3561, Jul. 2003.

[22] A. Caprara, H. Kellerer, and U. Pferschy, "The multiple subset sum problem," *SIJOP: SIAM J. Optimization*, vol. 11, pp. 308–319, 2000.

Stefano Avallone (A'06) received the M.S. degree in telecommunications engineering and the Ph.D. degree in computer networks from the University of Napoli "Federico II", Naples, Italy, in 2001 and 2005, respectively.

His research interests include computer networks, traffic engineering, QoS routing, wireless mesh networks. He was a visiting researcher at the Delft University of Technology (2003–2004) and at the Georgia Institute of Technology (2005).

In 2004, Dr. Avallone was awarded a research funding from the European Doctoral School of Advanced Topics in Networking (SATIN), the instrument employed by E-NEXT (an EU FP6 Network of Excellence) to invest in education of researchers for the European Research Area.

Ian F. Akyildiz (F'95) is a Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, and Director of the Broadband Wireless Networking Laboratory. His current research interests are in wireless mesh networks, cognitive radio networks, and sensor networks.

Dr. Akyildiz is the Editor-in-Chief of *Computer Networks* (Elsevier) and *Ad Hoc Networks* (Elsevier) journals. He served as a national lecturer for ACM from 1989 to 1998 and received the ACM Outstanding Distinguished Lecturer Award for 1994. He received the 1997 IEEE Leonard G. Abraham Prize award (IEEE Communications Society), the 2002 IEEE Harry M. Goode Memorial Award (IEEE Computer Society), the 2003 IEEE Best Tutorial Award (IEEE Communication Society), and the 2003 ACM SIGMOBILE award. He is an ACM fellow (1996).

Giorgio Ventre (M'95) is Professor of Computer Networks in the Department of Computer Engineering and Systems of the University of Napoli "Federico II". After starting ITEM, the first research laboratory of the Italian University Consortium for Informatics (CINI), he is now President and CEO of CRIAI, a research center active in the areas of Information Technologies. As leader of the networking research group at University of Napoli Federico II, he was principal investigator for a number of national and international research projects.

Dr. Ventre is Area Editor of *Computer Networks* (Elsevier). He is a member of the IEEE Computer Society and of the ACM.