A Traffic-Aware Channel and Rate Reassignment Algorithm for Wireless Mesh Networks

Stefano Avallone, *Member*, *IEEE*, Giovanni Di Stasi, and Andreas Kassler, *Member*, *IEEE*

Abstract—Channel assignment is among the most challenging issues for multiradio wireless mesh networks, given the variety of objectives that can be pursued and the computational complexity of the resulting problems. The channel assignment problem has been also shown to be interdependent with the routing problem, i.e., the problem to determine the amount of traffic flow to be routed on every link. Such a relationship raises the need to recompute the channel assignment every time the traffic pattern changes. However, channel assignment algorithms designed to assign channels from scratch will likely return a completely different configuration of radios, which would disrupt the network operation for the time required to switch to using the links established on the new channels. As shown by the experiments that we conducted, such a time may not be negligible, due to the resistance of routing protocols designed for wireless ad hoc and mesh networks to rapidly flagging a link as established/lost. Such a consideration, along with the observation that channel assignment algorithms may be suboptimal, led us to the design of a channel reassignment algorithm that takes the current channel assignment into account and attempts to cope with the new traffic pattern in the best manner possible while modifying the channel on a limited number of radios. In this paper, we illustrate such a channel reassignment algorithm and evaluate its performance by means of both simulations and experiments with real hardware.

Index Terms—Wireless mesh networks, channel reassignment, algorithm design and analysis

1 Introduction

Since their introduction, wireless mesh networks (WMNs) have attracted a lot of interest from both the international research community and industries. Such an interest from industries is due to the possibility to cover metropolitan areas without a wired infrastructure, which makes WMNs a cost-effective solution to implement, for instance, wireless ISPs. Researchers, instead, have been attracted by the challenging issues related to the configuration and management of WMNs. One of such issues is the assignment of channels to radios in case mesh routers are equipped with multiple radios. The multiradio configuration is becoming increasingly common, as routers may exploit the availability of multiple radios to simultaneously transmit and/or receive on different channels. Consequently, it is possible to reduce the interference and increase the throughput by carefully planning the assignment of channels to radios.

It has been shown [1] that the assignment of channels is not independent of the routing problem. Indeed, nodes using the same channel in a neighborhood have to share the channel capacity, and hence the amount of bandwidth available on a link depends on how many nodes are using the same channel in the neighborhood. Then, the way channels are assigned affects the amount of bandwidth available on links, and hence the channel assignment problem must be jointly studied with the routing problem. However, the joint channel assignment and routing problem has been shown to be NP-complete. Therefore, the proposals that recently appeared in the literature addressing such joint problem solve the channel assignment problem and the routing problem separately. A common approach is to first solve the routing problem, i.e., how to determine the amount of flow (referred to as the *flow rate*) to be routed on each link, and then to solve the channel assignment problem, i.e., how to assign channels in such a way that the resulting bandwidth available on each link exceeds the link flow rate.

Since the assignment of channels depends on the set of flow rates, it should be recomputed upon a variation of the traffic load. However, frequent recomputations of the channel assignment are not desirable. Indeed, a new execution of the channel assignment procedure does not take the current assignment into account and thus will likely return a completely different assignment of channels with respect to the current one. Enforcing the new assignment will thus require changing the channels assigned to several radios. Switching channel on a radio breaks the network connectivity for a much longer time than that required by the radio hardware to shift to the new frequency. Indeed, routing protocols take some time to assess that a previously active link is no longer available or a new link is actually reliable. That is necessary due to the varying conditions of the wireless medium and is done to avoid routing oscillations. Hence, when a radio is assigned a new channel,

- *S. Avallone and G. Di Stasi are with the Department of Computer Engineering, University of Naples, Via Claudio, 21, 80125 Napoli, Italy. E-mail: {stefano.avallone, giovanni.distasi}@unina.it.*
- *A. Kassler is with the Department of Computer Science, Karlstad University, Universitetgatan 2, 65188 Karlstad, Sweden. E-mail: andreas.kassler@kau.se, kassler@ieee.org.*

Manuscript received 19 July 2010; revised 15 Nov. 2011; accepted 19 Apr. 2012; published online 1 May 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2010-07-0343. Digital Object Identifier no. 10.1109/TMC.2012.107.

the routing protocol takes some time to start using the links established on the new channel instead of the links on the previous channel. The consequent packet losses may also induce the TCP entities to decrease the congestion window and increase the retransmission timer, thus lowering the throughput for an additional period. To support such statements, we conducted some experiments in the ORBIT testbed [2], which showed that a channel switch can break the network connectivity for up to 55 seconds.

Switching channel on a radio therefore results into pruning all the links using that radio from the network topology for a certain period of time. Thus, it is clear that the more radios switch channel, the higher the impact on the network performance. For this reason, we present a simple heuristic that takes the current channel assignment into account and aims to adjust at most a configurable number of channels to cope with a variation in the set of flow rates in the best manner possible. Through a thorough simulation study, we show that our heuristic, besides being beneficial in the short term due to the limited number of required channel switches, also ensures a higher throughput in the longer term, with respect to both other channel assignment algorithms and the strategy of leaving the channel assignment unchanged. Indeed, as the channel assignment problem is NP-complete [1], most existing algorithms are heuristics that only provide a suboptimal solution. Our channel reassignment algorithm, instead, starts from one such solution and makes some adjustments to find a better solution.

Another contribution of this paper is an analysis of the overhead introduced by the IEEE 802.11 medium access function that led to the identification of a parameter, the *total utilization* of a collision domain, that has a strong (negative) correlation with the network throughput. Hence, we considered the minimization of the maximum total utilization over all the collision domains as the objective of our channel reassignment algorithm. Also, the aforementioned analysis enables us to find a *reference* value for the maximum total utilization that ensures that the network is actually able to carry the offered traffic load. Thus, the condition that the maximum total utilization exceeds such reference value can be used as an indication that a channel reassignment is needed. We applied such criterium in a simulation conducted with real traffic traces, which showed that reassigning channels by using our heuristic allows a remarkable throughput increase with respect to the strategy of leaving the channel assignment unchanged.

The rest of this paper is organized as follows: Section 2 gives an overview of the related work. Section 3 formalizes the system model and the channel reassignment problem. In Section 4, we illustrate the operation of the proposed algorithm by means of its pseudocode. In Section 5, we show the results of the simulation studies and the experiments carried out to evaluate the performance of our algorithm. Finally, Section 6 concludes our paper.

2 Related Work

The channel assignment problem in multiradio 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, in [3], the goal is to find a channel assignment that minimizes the size of the largest collision domain subject to the constraint that the induced graph must still be K-connected. A centralized algorithm is presented in [4] which also takes the traffic generated by mesh clients into account. In [5], an interference-free channel assignment is sought by using superimposed codes. MesTiC [6] is a rank-based channel assignment, where the rank of a node is a function of its aggregate traffic, its number of hops from the gateway and its number of radio interfaces. In [7], both centralized and distributed algorithms are presented, which aim to minimize the number of pairs of links that are interfering. A distributed channel assignment algorithm and a distributed routing protocol are proposed in [8]. Dhananjay et al. [9] present a distributed protocol for channel assignment and routing in dual-radio mesh networks.

Other proposals study the joint channel assignment and routing problem. An iterative routing algorithm based on traffic profiles is proposed in [10]. In [11], an approximate solution for the joint channel assignment and routing problem is developed which optimizes the network throughput subject to fairness constraints. The problem how to verify the feasibility of a given set of flows between source-destination pairs is investigated in [12]. In [13], a distributed joint channel assignment, scheduling, and routing algorithm is presented. In [1], tuning the transmission rate is exploited to present a channel and rate assignment heuristic. In [14], the authors develop a centralized solution to the joint logical topology design, interface assignment, channel allocation and routing problem. For a more comprehensive survey of channel assignment algorithms for WMNs, we refer the reader to [15]. There has been also some work on the channel assignment problem in wireless sensor networks. Due to hardware limitations, sensors have a single radio interface, and thus the proposed algorithms usually assign channels in a dynamic manner. A traffic-aware channel assignment algorithm is proposed in [16], while in [17] a middleware positioned between the MAC and PHY layers is proposed to find the best channel at runtime and communicate it to a single-channel MAC protocol. A protocol to detect the radio interference among nodes and a collision-free TDMA schedule based on the results of such detection are proposed in [18].

All the papers mentioned so far, however, do not consider the problem how to reconfigure the WMN after a change in the traffic flows. Such a problem has been tackled in a few papers. The approach in [19] does not consider the standard CSMA/CA access technique but assumes the existence of a link layer synchronization among the nodes which enables them to organize their data transmissions in different time slots with no contention. Hence, the proposed algorithm reconfigures the channel assignment and the link scheduling as a consequence of a change in the traffic matrix. Our approach, instead, assumes the standard contention-based access technique and hence does not perform link scheduling. In [20], a distributed channel reassignment heuristic is proposed that aims to cope with the traffic variation due to mesh clients handoffs. The proposed approach solely makes a channel switch on the edge mesh routers aggregating clients traffic and does not ensure that all the node pairs remain connected after the channel switch, thus requiring changes in the routing tables of the mesh routers involved. With respect to our previous approach [21], this paper presents an enhanced version of the channel reassignment algorithm under many aspects (reconfiguration of transmission rates, improved definition of the link priorities, and so on) and a more accurate and complete performance evaluation.

3 Problem Formulation

3.1 System Model

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 $\{r_m\}_{m=1}^{M}$. We adopt the *physical* model of interference, which considers a transmission successful if the signal-to-interference and noise ratio (SINR) at the receiver is sufficiently high to decode the signal. The SINR at receiver v when a signal is transmitted by u is defined as

$$SINR_{uv} = \frac{G_{uv} P_u}{\sum_{x \to y \neq u \to v} G_{xv} P_x + n_v},$$

where P_u is the power emitted by u to transmit to v, 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 $SINR_{uv} \geq \gamma_{r_m}$, where γ_{r_m} denotes the minimum SINR required to correctly decode a signal modulated at the rate r_m.

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, in the absence of transmissions on other links, the signal to noise ratio at v is larger than the SINR threshold for one of the available transmission rates r_m, i.e., $\frac{G_{uv} P_u}{n_v} \geq \gamma_{r_m}$. The capacity of the link $c(u \to v)$ is then set to the highest transmission rate for which the signal to noise ratio is larger than the corresponding threshold. An edge $u \to v \in E_I$ indicates that u can transmit to v provided that they are assigned a common channel. 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 $u \to v \in E_I$ *and* they share at least one common channel. 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 $u \stackrel{c}{\to} v$. Finally, we say that a link $x \stackrel{c}{\to} y \in E$ interferes with $u \stackrel{c}{\to} v \in E$ if a simultaneous transmission on $x \stackrel{c}{\to} y$ prevents v from correctly decoding the signal from u.

3.2 A Condition on the Flow Rates of Interfering Links

The effect of the interference is to prevent simultaneous transmissions over neighboring links using the same channel. Hence, the throughput that can be achieved across a wireless link (denoted as *flow rate* in the following) is affected by the amount of traffic transmitted on the neighboring links. Given a set of links L such that no two links can be transmitting simultaneously, our goal is to determine a condition establishing whether the associated flow rates can be actually achieved or not. In a given time interval of duration T, each link $e \in L$ with a flow rate $f(e)$ has to carry an amount of data equal to $f(e)T$. If we denote by p the average size of the frame body, such amount of data is transmitted by means of $\frac{f(e)}{p}T$ packets. The time required to transmit a packet is given by the time actually needed to transmit the frame body $(\frac{p}{c(e)})$ plus the overhead introduced by the medium access function (denoted as $\Omega(e)$). Since no two links of the set L can be transmitting simultaneously, a necessary condition for the associated set of flow rates to be achievable is that the sum of the amount of time required by every link to transmit the necessary packets to guarantee the corresponding flow rate be less than T, i.e., $\sum_{e \in L} (\frac{p}{c(e)} + \Omega(e)) \cdot \frac{f(e)}{p} T \leq T$, which yields

$$\sum_{e \in L} \frac{f(e)}{c(e)} \leq 1 - \sum_{e \in L} \frac{f(e)}{p} \Omega(e), \tag{1}$$

where the summation in the right-hand side (RHS) represents the sum over all the links in L of the overhead related to the transmission of a packet times the number of packets sent per second. The RHS of (1) thus represents an upper bound to the sum of the flow to capacity ratios that can be actually achieved. To derive a tighter upper bound, transmission failures might be taken into account by multiplying the number of packets sent in the interval of duration T by the average number of transmission attempts. However, we are interested in determining the highest value possible for the sum of the flow to capacity ratios, and thus we consider the ideal case of the absence of collisions.

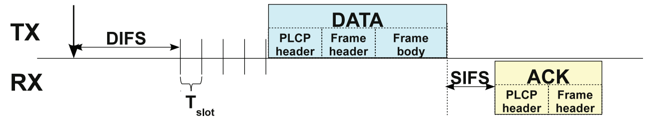


Fig. 1. Wireless medium access according to 802.11 DCF.

We now show how to evaluate $\Omega(e)$ in case the basic (i.e., without advanced features such as block ack, transmission opportunity, frame aggregation) 802.11 distributed coordination function (DCF) is used to access the wireless medium. As shown in Fig. 1, all the time intervals but the time required to send the frame body are to be considered as overhead:

$$\Omega(e) = DIFS + T_{slot} \cdot N_{slot} + 2 \cdot T_{PLCP} + \frac{HLEN}{c(e)} + SIFS + \frac{ACK}{R_{ctrl}},$$

where $HLEN$ is the size of the 802.11 header, T_{PLCP} is the time to transmit the physical layer convergence procedure (PLCP) preamble, ACK is the size of the ack frame, and R_{ctrl} is the rate used to transmit control frames. Given the assumption that no transmission failure occurs, the contention window of all the stations stays at its minimum value (CW_{min}) and the number of slots (N_{slot}) a station waits in

Parameter	Value
$SIFS$	$16\mu s$
T_{slot}	$9\ \mu s$
$DIFS$	$SIFS + 2{\cdot}T_{slot}$
CW_{min}	15
T_{PLCP}	$23\ \mu s$
$HLEN$	$28B$
ACK	$14B$
R_{ctrl}	$6\ Mbps$

Fig. 2. 802.11a parameters.

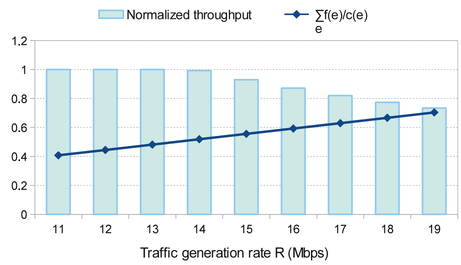


Fig. 3. ns-3 simulation (UDP traffic).

the backoff stage is, on the average, half the value of the contention window.

An upper bound to the sum of the flow to capacity ratios can be easily determined in case all the links in the set L use the same transmission rate (denoted as c). In such a case, $\Omega(e) = \Omega$ and (1) yields

$$\frac{\sum_{e\in L} f(e)}{c} \leq \frac{p}{p + \Omega c}. \quad (2)$$

If we consider the physical layer specified in 802.11a (whose specific values are reported in Fig. 2), $c = 54$ Mbps and an average packet payload size of $1{,}400B$ (which leads to a frame body size of $1{,}428B$ if we consider UDP and IP headers), then the RHS of (2) is approximately equal to 0.53. Thus, in the considered case, the maximum value for the sum of the flow rates on all the links is about half the physical transmission rate. We performed a simple experiment using the network simulator ns-3 to support this result. Two nodes having a single radio are placed at a distance of 30 m (so that they transmit at 54 Mbps) and each of them generates UDP traffic at rate R destined to the other node. According to our analysis, the generated traffic rates are achievable if the sum of the flow to capacity ratios is below 0.53, i.e., $2R/54 \leq 0.53$, which implies $R \leq 14.3$ Mbps. We vary the rate R from 11 to 19 Mbps and measure the average throughput over 30 seconds. The bars in Fig. 3 show the measured throughput normalized to the rate of the generated traffic ($2R$), while the symbols indicate the sum of the flow to capacity ratios ($2R/54$). The figure shows that the measured throughput equals the generated traffic rate until $R = 14$ Mbps. If we increase the generated traffic rate R beyond the maximum value determined based on our analysis (14.3 Mbps), the throughput reaches a saturation value ($\simeq 28$ Mbps) and then drops more and more below the generated traffic. Such experiment thus confirms that a set of flow rates are actually achievable if the sum of the flow to capacity ratios is below a certain threshold. Beyond such threshold, the achieved throughput is a decreasing fraction of the offered load. Also, the experiment shows that the threshold value has been correctly predicted by our analysis.

In case of TCP traffic, we also need to consider the TCP acknowledgments. Assuming that TCP acks traverse the same nodes as the TCP segments (in the opposite direction) and are not piggybacked by the TCP segments in the reverse direction, we need to consider on each link e of the set L an additional flow rate of $\frac{f(e)}{p} p_{ack}$ (i.e., the number of TCP acks per second is the same as the number of TCP segments sent per second), where $p_{ack} = 40B$. Thus, an upper bound to the sum of the flow to capacity ratios, in case all the links use the same transmission rate and transmit TCP traffic, can be derived from (1) as well:

$$\frac{\sum_{e\in L} f(e)}{c} \leq \frac{p}{p + p_{ack} + 2\Omega c}. \quad (3)$$

Considering again the 802.11a physical layer, $c = 54$ Mbps and an average packet payload size of $1{,}400B$ (which leads to a frame body size of $1{,}440B$ if we consider TCP and IP headers), we obtain from (3) that the maximum sum of the flow to capacity ratios is about 0.37. We perform a similar ns-3 experiment to validate such result. According to our analysis, we expect that the maximum traffic generation rate for each TCP source is $0.37 \cdot 54/2 \simeq 10$ Mbps. We vary the traffic generation rate R from 7 to 15 Mbps and measure the average throughput over 30 seconds. The results (Fig. 4) show that the throughput is equal to the generated traffic rate until R is lower than or equal to 11 Mbps (or, equivalently, the sum of the flow to capacity ratios is lower than 0.40). The fact that the bounds provided by our analysis are exceeded can be likely explained by considering that a TCP ack is not necessarily sent for every single TCP segment, but the transmission of a TCP ack can be delayed to acknowledge more than one segment. Hence, the rate of the TCP acks is lower and there is room for sending more TCP segments. Nevertheless, these simulations show that the maximum achievable sum of the flow to capacity ratios is bounded and (3) can provide a good approximation of such bound.

3.3 The Total Utilization of a Collision Domain

In the previous section, we derived (1) as a necessary condition for a given set of flow rates associated with links interfering with each other to be actually achieved. To apply such condition to a network topology, we need to determine all the sets of links such that no two links in a set can be transmitting simultaneously. For this purpose, we might build a *conflict graph* [22], i.e., a graph where vertices

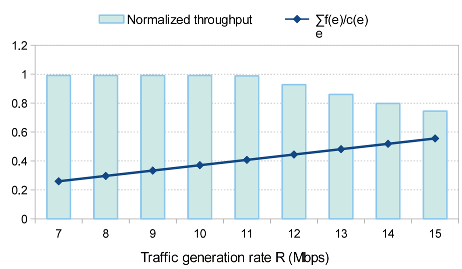


Fig. 4. ns-3 simulation (TCP traffic).

represent network links and edges connect vertices representing interfering links, and find all the maximal cliques in the conflict graph. However, finding all the maximal cliques in a graph is a known NP-complete problem. To lower the complexity, we consider the notion of *collision domain* of a link.

We define the collision domain of a link $u \stackrel{c}{\rightarrow} v$ as the set of all the links that interfere with it. Formally,

$$\mathcal{D}(u \stackrel{c}{\rightarrow} v) = \left\{ x \stackrel{c}{\rightarrow} y \in E \mid \frac{G_{uv}P(u \rightarrow v)}{G_{xv}P(x \rightarrow y) + n_v} < \gamma_{c(u \stackrel{c}{\rightarrow} v)} \right\} \cup \{ v \stackrel{c}{\rightarrow} u \}.$$

Thus, by definition, $u \stackrel{c}{\rightarrow} v \in \mathcal{D}(u \stackrel{c}{\rightarrow} v)$. Also, $v \stackrel{c}{\rightarrow} u$ belongs to $\mathcal{D}(u \stackrel{c}{\rightarrow} v)$ as a single radio cannot transmit and receive simultaneously. Thus, none of the links in the collision domain of link e can be active at the same time as e. However, two links in the collision domain of e might be able to transmit simultaneously. It follows that, when applied to the links of a collision domain, (1) is no longer a necessary condition for achieving the set of flow rates. Hence, the sum of the flow to capacity ratios over the links of a collision domain can exceed the RHS of (1). Also, unlike the previous experiments, links usually have different capacities, and hence it is not easy to derive an upper bound to the sum of the flow to capacity ratios from (1).

Nonetheless, we still strive to ensure that the sum of the flow to capacity ratios over the links of a collision domain is below a certain threshold, for every collision domain. For conciseness, we define the sum of the flow to capacity ratios over the links of the collision domain of link e as the *total utilization* of that collision domain and denote it by $U_{tot}(e) = \sum_{e_0 \in \mathcal{D}(e)} \frac{f(e_0)}{c(e_0)}$. Our objective to have the total utilization of all the collision domains below a threshold (which is equivalent to have the maximum among the total utilizations of all the collision domains below that threshold) is supported by the experiments in Section 5.1.5, which show that the achieved throughput is very close to the offered load as long as the maximum total utilization over all the collision domains is below a certain threshold. Henceforth, by *maximum total utilization*, we mean the maximum among the total utilizations of all the collision domains.

3.4 The Channel Reassignment Problem

Given the values of flow rate f for every link $e \in E$ and a channel assignment, the channel reassignment problem is to change the channels assigned to at most a given number of radios so that the total utilization of every collision domain (or, equivalently, the maximum total utilization) is below a given threshold, and the network topology is preserved (meaning that there must be a link between every two nodes that were connected before the channel reassignment). Being equivalent to the channel assignment problem, but with the additional constraint on the number of radio changes, the channel reassignment problem is NP-complete, too [1]. Hence, it is not possible to determine in polynomial time whether a solution to the channel reassignment problem exists for a given threshold. Consequently, the heuristic we propose aims to minimize the maximum total utilization over all the collision domains while changing the channels assigned to at most the given number of radios.

```
MVCRA-R(G(V,E), {f(e)}_{e∈E}, MaxNumChanges, λ0)
U_tot(e) ← Σ_{e0∈D(e)} f(e0)/c(e0)   ∀e ∈ E
Q ← {e}_{e∈E}
Num_Changes ← 0
while Q ≠ ∅ AND (Num_Changes < MaxNumChanges)
      do (u -c_old-> v) ← EXTRACT_MAX(Q)
         (c_sel, c(u -c_sel-> v)) ← MIN_UTOT(u, v, C)
         CHANGE_IF(u, c_sel)
         CHANGE_IF(v, c_sel)
         E ← E − {u -c_old-> v} ∪ {u -c_sel-> v}
         while Q_P is not empty
              do (s → t) ← EXTRACT_MAX(Q_P)
                 S ← A(s) ∩ A(t)
                 if S = ∅
                    then if Σ_{k∈C} count_s(k) > Σ_{k∈C} count_t(k)
                            then S ← A(s)
                            else S ← A(t)
                 (c_sel, c(s -c_sel-> t)) ← MIN_UTOT(s, t, S)
                 CHANGE_IF(s, c_sel)
                 CHANGE_IF(t, c_sel)
                 E ← E − {s -c_old-> t} ∪ {s -c_sel-> t}
```

Fig. 5. Pseudocode MVCRA-R.

In the attempt to minimize the total utilization, we also exploit the availability of multiple transmission rates. Indeed, the total utilization $U_{tot}(e) = \sum_{e_0 \in \mathcal{D}(e)} \frac{f(e_0)}{c(e_0)}$ 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 possible for all the links 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, $\mathcal{D}(e|c(e) = r_i) \subseteq \mathcal{D}(e|c(e) = r_j)$ for $i < j$. Thus, decreasing the transmission rate on link e may help reduce the total utilization of its collision domain. Also, 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. Thus, our proposed channel reassignment algorithm, presented in the next section, adjusts both the channel and the transmission rate on selected links to further decrease the maximum total utilization.

4 Minimum Variation Channel and Rate Reassignment Algorithm (MVCRA-R)

In this section, we present the MVCRA-R algorithm. We illustrate the operation of our algorithm through the pseudocode shown in Figs. 5, 6, 7, 8, and 9. MVCRA-R is given the current assignment of channels, the new set $f(e)$ of flow rates and the *MaxNumChanges* parameter, which determines the maximum allowed number of changes to the channels assigned to the radios. To determine what radios

MIN_UTOT$(u, v, \mathcal{S})$
1 **for** each $c \in \mathcal{S}$
2 **do** $U'_{max}(c) \leftarrow \max_{x \xrightarrow{c} y \,|\, u \xrightarrow{c} v \in D(x \xrightarrow{c} y)} U_{tot}(x \xrightarrow{c} y)$
3 $r_c \leftarrow$ ADJUST_RATE$(u \xrightarrow{c} v, U'_{max}(c))$
4 $U_{max}(c) \leftarrow \max(U'_{max}(c), U_{tot}(u \xrightarrow{c} v))$
5 **return** $\left(\operatorname{argmin}_{c \in \mathcal{S}} U_{max}(c),\, r_c\right)$

Fig. 6. Pseudocode MIN_UTOT.

need to be assigned a new channel, we first compute the total utilization of all the collision domains as determined by the current channel assignment and the new set of flow rates (line 1). All the links of the communication graph are then inserted into a priority queue Q and are extracted one by one (line 4) in decreasing order of priority. The priority of a link l is given by its flow to capacity ratio times the number of links whose collision domain includes l and has a total utilization above a given threshold λ_0. The rationale is that we want to extract first those links that allow as many collision domains as possible to benefit from a channel switch. The $Num\_Changes$ variable holds the current number of channel adjustments and should not exceed the $MaxNumChanges$ parameter.

When a link $u \rightarrow v$ is extracted (we denote by c_{old} the channel it is currently assigned), the goal is to determine a new channel c (independently from the channels currently assigned to u and v), and a new rate r, that minimize the total utilization of its collision domain (line 5). This is achieved by invoking the MIN_UTOT function (Fig. 6), which analyzes the effects of assigning each of the potential channels to link $u \rightarrow v$ and returns the most convenient one. In particular, in order not to take decisions that might aggravate the total utilization of other collision domains, the MIN_UTOT function also considers, for each channel c in the set $\mathcal{S}$, the total utilization of the collision domain of all the links $x \xrightarrow{c} y$ which would have $u \xrightarrow{c} v$ in their collision domain. In case a link $u \xrightarrow{c} v$ was established, all such total utilizations would be increased by the same amount, i.e., $\frac{f(u \xrightarrow{c} v)}{c(u \xrightarrow{c} v)}$. To keep track of the effects of assigning a channel c to the extracted link on such collision domains, it suffices to only consider the maximum among such total utilizations, which is denoted by $U'_{max}(c)$ (line 2 in Fig. 6). If a link $u \xrightarrow{c} v$ were established, we would also need to consider the total utilization of its collision domain. $U_{tot}(u \xrightarrow{c} v)$ may be decreased by reducing the transmission

CHANGE_IF(u, c)
1 **if** $(c \in \mathcal{A}(u)$ or $|\mathcal{A}(u)| < k(u))$
2 **then** return
3 $(k, W_k) \leftarrow$ MIN_DISRUPT(u, c)
4 $Q_P \leftarrow Q_P \cup W_k$
5 $Q \leftarrow Q - W_k$
6 $\mathcal{A}(u) \leftarrow \mathcal{A}(u) - \{k\} \cup \{c\}$
7 $count_u(c) + +$
8 $\overline{W}_k \leftarrow \left\{u \xrightarrow{k} w \in E \vee w \xrightarrow{k} u \in E\right\} - W_k$
9 **for** each $x \xrightarrow{k} y \in \overline{W}_k$
10 **do** $\mathcal{S} \leftarrow \mathcal{A}(x) \cap \mathcal{A}(y)$
11 $\left(c_{sel},\, c(x \xrightarrow{c_{sel}} y)\right) \leftarrow$ MIN_UTOT$(x, y, \mathcal{S})$
12 $E \leftarrow E - \{x \xrightarrow{k} y\} \cup \{x \xrightarrow{c_{sel}} y\}$
13 $Num\_Changes + +$

Fig. 8. Pseudocode CHANGE_IF.

rate on $u \xrightarrow{c} v$, because a lower rate may allow to reduce the size of the collision domain. For the purpose of determining the most suitable rate, the ADJUST_RATE function (Fig. 7) is invoked. Such a function starts by considering the highest rate possible and then proceeds by iteratively trying lower rates, as long as $U_{tot}(u \xrightarrow{c} v)$ is greater than $U'_{max}(c)$. We note that the rate is actually decreased only if it allows to reduce the total utilization $U_{tot}(u \xrightarrow{c} v)$ (line 5 in Fig. 7). Then, the MIN_UTOT function computes the collision domain of $u \xrightarrow{c} v$ considering the rate returned by ADJUST_RATE and uses the $U_{max}(c)$ variable to hold the maximum between $U_{tot}(u \xrightarrow{c} v)$ and $U'_{max}(c)$. MIN_UTOT returns the channel that minimizes $U_{max}(c)$ and the rate selected for that channel.

Since the channel c_{sel} returned by MIN_UTOT may not be currently assigned to any radio on u and v, the CHANGE_IF function is invoked (lines 7 and 8 in Fig. 8) to set a radio interface of nodes u and v to c_{sel}. The CHANGE_IF function (Fig. 8) is passed the node u and the channel c that has to be assigned to one of u's radios. If channel c is already assigned to one of u's radios or u has an available radio, then nothing else needs to be done (lines 1 and 2). Otherwise, we attempt to assign channel c to the radio of u that causes the least disruption in the network configuration. For this purpose, the MIN_DISRUPT function is invoked (Fig. 9), which computes, for every channel k currently assigned to node u, the set W_k of links that would be disrupted by switching a radio on u from channel k to c. Clearly, all the links between u and the nodes that still share a common channel with u after the channel switch

ADJUST_RATE(e, U_{max})
1 $m \leftarrow \max\{i \in 1 \ldots M \,|\, e \in E_I \wedge c(e) = r_i\}$
2 $m_{min} \leftarrow m,\ min \leftarrow U_{tot}(e)$
3 **while** $m > 1$ AND $U_{tot}(e) > U_{max}$
4 **do** $m \leftarrow m - 1,\ c(e) \leftarrow r_m$
5 **if** $U_{tot}(e) < min$
6 **then** $min \leftarrow U_{tot}(e)$
7 $m_{min} \leftarrow m$
8 **return** $r_{m_{min}}$

Fig. 7. Pseudocode ADJUST_RATE.

MIN_DISRUPT(u, c)
1 $W_k \leftarrow \{u \xrightarrow{k} w \in E \vee w \xrightarrow{k} u \in E \,|\, (\mathcal{A}(u) - \{k\} \cup \{c\}) \cap \mathcal{A}(w) = \emptyset\} \qquad k \in \mathcal{A}(u)$
2 $\omega_k \leftarrow \left(1 + \frac{count_u(k)}{\sum_{k_0 \in \mathcal{C}} count_u(k_0)}\right) \cdot \sum_{l \in W_k} \frac{f(l)}{c(l)} \qquad k \in \mathcal{A}(u)$
3 **return** $\left(\operatorname{argmin}_{k \in \mathcal{A}(u)} \omega_k,\, W_k\right)$

Fig. 9. Pseudocode MIN_DISRUPT.

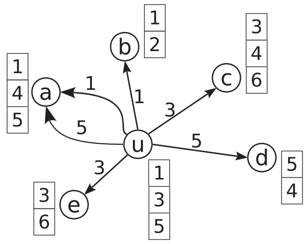


Fig. 10. Example to illustrate MIN_DISRUPT.

can be easily fixed by using one of the common channels. That happens when a neighbor of u has a radio on channel c or u and its neighbor share more than one channel before the channel switch. As an example, Fig. 10 illustrates the case where channel 2 has to be assigned to one of the radios on u. In such example (where, for simplicity, links entering u are not shown), $\mathcal{A}(u) = \{1, 3, 5\}$ and $W_1 = \emptyset$ (because $2 \in \mathcal{A}(b)$ and $5 \in \mathcal{A}(u) \cap \mathcal{A}(a)$), $W_3 = \{u \xrightarrow{3} c,\ u \xrightarrow{3} e\}$ and $W_5 = \{u \xrightarrow{5} d\}$ (because $1 \in \mathcal{A}(u) \cap \mathcal{A}(a)$). For each channel k currently assigned to node u, a weight ω_k is computed (line 2), which is composed of two factors. The first factor is a function of the (normalized) number of times channel k has been assigned (by CHANGE_IF) to node u. Thus, the weight of the channels that have been previously assigned to u is higher to make it less likely that they are replaced by new channels. The second factor is the sum of the flow to capacity ratio of all the links that would be disrupted by replacing channel k with c on node u. Such a factor accounts not only for the number of links that would be disrupted, but also for the amount of flow they carry. Minimizing the number of links to be repaired as a consequence of a channel switching is important to meet the constraint on the maximum allowed number of radio changes. Accounting for the amount of flow on the pending links is important as well, because a pending link might be assigned a channel that minimizes further disruptions rather than one that minimizes the maximum total utilization over all the collision domains that include it. Hence, it is preferable to disrupt links carrying a lower amount of flow to minimize the impact on the total utilization of the collision domains that will include the pending links.

MIN_DISRUPT returns the channel k with the minimum weight ω_k, which then has to be replaced by c on node u. Consequently, all the links on u that were using channel k must be assigned a new channel. To do so, CHANGE_IF inserts all the links belonging to W_k into the queue Q_P of the *pending* links, i.e., links that need additional channel switches to be repaired (line 4 in Fig. 8). Such links are also removed from the queue Q, because they will be processed when extracted from the queue Q_P. Channel c replaces channel k on node u and the counter of the number of times that channel c has been assigned to u is increased by one (lines 6-7). Then, all the links that were using channel k and are not in W_k are fixed by being assigned a new channel. For this purpose, MIN_UTOT is called to determine the common channel between the two end nodes that minimizes the impact on the total utilization of the other collision domains. Finally, $Num\_Changes$ is increased to reflect the channel adjustment on u and CHANGE_IF returns.

As mentioned above, a call to CHANGE_IF from MVCRA-R (lines 7 and 8 in Fig. 8) may bring some links into a pending state, where they need to be assigned a new channel. MVCRA-R will thus extract the links from Q_P one by one, in decreasing order of priority, until the queue is empty (line 11). The priority of a pending link is its flow to capacity ratio. The goal is to extract (and repair) first those links that will contribute more to the total utilization of the collision domains in which they will be included. For each extracted link $s \rightarrow t$, the set $\mathcal{S}$ of the common channels between s and t is determined. If $\mathcal{S}$ is empty, we necessarily need to change a channel on either s or t. In such a case, $\mathcal{S}$ is filled with the channels of the node that has experienced the highest number of channel switches (line 14), so that a channel is changed on the other node. The MIN_UTOT procedure is invoked to determine the channel of $\mathcal{S}$ and the rate minimizing the resulting maximum total utilization. Then, CHANGE_IF is called to actually assign the selected channel to one of the radios on s and t. Clearly, one or both of these calls (depending on whether or not s and t shared a common channel) return immediately, because one or both of the end nodes already have a radio on the selected channel. Finally, the extracted pending link is switched to the selected channel (line 20). When MVCRA-R ends, the queue of the pending links is empty, thus ensuring that all the links have been assigned a channel and hence the network topology is preserved.

5 Performance Evaluation

In this section, we present the results of the simulation study and the experiments we carried out to evaluate the performance of MVCRA-R algorithm. The goal of the simulation study is to show that updating the network configuration by running MVCRA-R allows to increase the network throughput with respect to both leaving the channel assignment unchanged and updating the network configuration by running a channel assignment algorithm that ignores the current configuration. We remark that we do not simulate the transient stage when radios switch channels (results may be affected by inaccuracies in the simulator). Instead, simulations start from the new network configuration determined by the channel (re-)assignment algorithms. Thus, simulations aim at evaluating the throughput in the long term.

The experiments that we conducted with real hardware, instead, aim to gain some insight into the effects of switching channels. We show that a large number of simultaneous channel switches severely impacts the network performance, thus justifying our objective of limiting the number of channel switches.

5.1 Simulation Study

We conducted a simulation study to compare MVCRA-R to MVCRA [21] and FCRA (Flow-Based Channel and Rate Assignment) [1] in terms of maximum total utilization, number of radios that have to switch channel and average network throughput. FCRA is a greedy channel assignment heuristic that extracts all the links one-by-one and assigns each link the channel that currently minimizes the maximum total utilization among all the collision domains including the extracted link.

TABLE 1
Network Topologies Characteristics

Topology	Nodes	Radios	Average node degree ($\|E\|/\|V\|$)	Area (m^2)
A	22	57	4.36	125×155
B	22	57	4.54	185×235
C	28	75	5.35	195×210

5.1.1 *Simulation Setup*

We consider three network topologies (whose main parameters are reported in Table 1) where each node is equipped with two or three radios. Given the planar coordinates of the nodes, a software that we implemented on our own is used to build the network topology based on the interference model described in Section 3. We assume that the gain G_{uv} of the radio channel between u and v to be the reciprocal of the square of the distance between u and v and the thermal noise to be -20 dbm. The SINR thresholds are set to allow a rate of 54 Mbps when the nodes are within 30 m, 48 Mbps within 32 m, 36 Mbps within 37 m, 24 Mbps within 45 m, 18 Mbps within 60 m, 12 Mbps within 69 m, 9 Mbps within 77 m, and 6 Mbps within 90 m. We assume that six nonoverlapping channels are available.

The initial set of flow rates is determined as follows: A subset of mesh nodes is identified as source or destination of traffic flows. Each source-destination pair is associated with a demand, whose amount of traffic is initially determined according to a random variable, as specified below. Each traffic demand is routed along either the shortest path or the three shortest paths [23] between the source and the destination. The sum of the amount of traffic routed on a link over all the source-destination pairs determines the flow rate on that link. FCRA is then used to compute the initial channel assignment. After a variation in the traffic demands, the way flows are routed is not changed. Hence, the share of a flow rate associated with a traffic demand is multiplied by the ratio of the new amount to the previous amount of that traffic demand.

We consider two types of traffic variation, denoted as "Increase" and "Swap." There are eight traffic demands in both cases. In the Increase strategy, the initial amount of traffic (denoted by L) is the same for all the demands and then the amount of traffic of each demand is scaled by a factor derived from a uniform distribution $U(0.5 + \alpha, 2\mu - 0.5 - \alpha)$. The mean value of such distribution is μ, which is chosen such that $\mu L = 4$, i.e., the average amount of traffic of a demand after the variation is 4 Mbps in all the cases. We performed simulations for $L \in \{1.5, 2, 2.5\}$ and $\alpha \in \{0, 0.1, 0.2, 0.3\}$. In the Swap strategy, the initial amount of traffic of each demand is derived from a probability distribution. We considered 10 different distributions in total: $U(1,5)$, $U(1,6)$, $U(1,7)$ and mixtures (with different probabilities) of two uniform distributions such as $U(1,2)$ and $U(3,4)$ or $U(1,2)$ and $U(5,6)$. Then, the amount of traffic of all the demands are swapped, in the sense that the demand with the highest amount of traffic gets the minimum amount of traffic, the demand with the second highest amount of traffic gets the second minimum amount of traffic, and so on. The network throughput is measured by means of simulations conducted with the *ns-3* network simulator. The physical layer specified in IEEE 802.11a is used for all the simulations. Each simulation lasts 60 seconds.

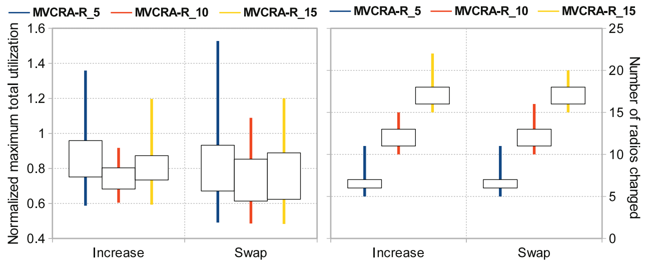


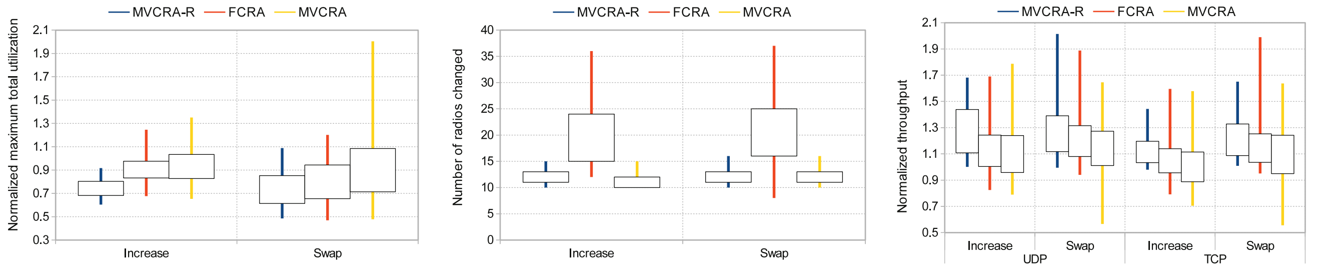
Fig. 11. Performance of MVCRA-R with different values of *MaxNumChanges*.

5.1.2 *Performance of MVCRA-R with Different Values of MaxNumChanges*

This set of simulations aims to evaluate the performance of MVCRA-R with different values of *MaxNumChanges*. The traffic variations described in the previous section (Increase and Swap) have been simulated in each of the three topologies considered and for each of the two strategies to route the initial traffic demands. MVCRA-R has been used to compute the new channel assignment starting from the initial channel assignment and the set of flow rates as of after the traffic variation.

Fig. 11 (left side) shows the distribution of the maximum total utilization achieved in all the simulations (for each of the two types of traffic variation) normalized to the maximum total utilization resulting from leaving the channel assignment unchanged. Thus, a normalized maximum total utilization below 1 means that reassigning the channels enabled a reduction in the maximum total utilization. A vertical line spans from the minimum to the maximum value over all the simulations, while a white box spans from the first quartile to the third quartile. Fig. 11 (right side) shows the distribution of the number of radios actually changed by MVCRA-R in all the simulations.

The performance of MVCRA-R has been evaluated for *MaxNumChanges* ranging from 5 to 15, though Fig. 11 only shows the results for three representative values (5, 10, 15). For all the values that we tested, we observed that MVCRA-R is able to meet, with a good approximation, the constraint on the maximum allowed number of radio changes. Regarding the maximum total utilization, we can observe (left side of Fig. 11) that the performance of MVCRA-R with $MaxNumChanges = 5$ is rather poor (in some cases the achieved maximum total utilization is more than 20 percent higher than that obtained by leaving the channel assignment unchanged). Performance tends to improve as *MaxNumChanges* increases and achieves the optimum for a value of 10. Increasing *MaxNumChanges* further does not bring any benefit. Thus, we implicitly assume in the following that MVCRA-R is used with $MaxNumChanges = 10$, which provides the best performance with the considered topologies and traffic loads.



(a) Normalized max total utilization (b) Number of radios changed (c) Normalized throughput

Fig. 12. Single traffic variation.

5.1.3 Single Variation in the Traffic Demands

Our goal is to evaluate the gain achieved by different channel assignment algorithms with respect to the strategy of leaving the channel assignment unchanged when a (single) variation in the traffic demands occurs. The traffic variations described in the previous section (Increase and Swap) have been simulated in each of the three topologies considered and for each of the two strategies to route the initial traffic demands. We evaluate how MVCRA-R, FCRA, and MVCRA react to each such traffic variations by feeding them with the previous channel assignment (ignored by FCRA) and the set of flow rates as of after the traffic variation.

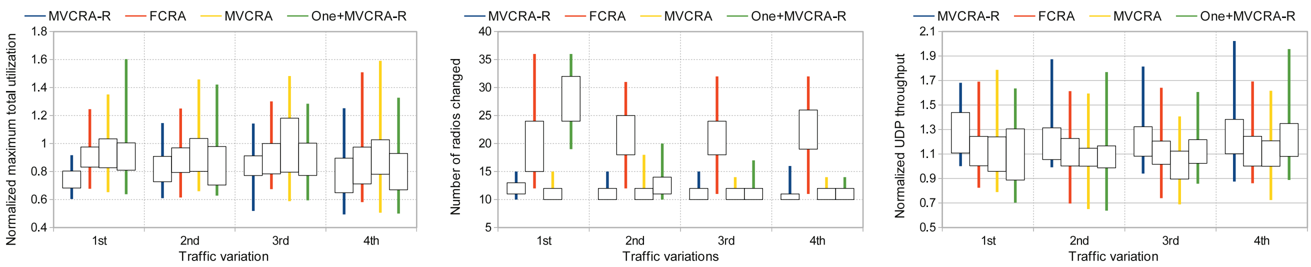
Fig. 12a shows the distribution of the maximum total utilization achieved by each algorithm normalized to the maximum total utilization resulting from leaving the channel assignment unchanged. The best performance is achieved by MVCRA-R, which achieves a 25 percent (on the average) reduction in the maximum total utilization with respect to the strategy of leaving the channel assignment unchanged, both in the Increase and in the Swap cases. FCRA achieves, on the average, a 10 percent (20 percent) reduction in the Increase (Swap) case, while MVCRA achieves a 7 percent (10 percent) reduction in the Increase (Swap) case.

Fig. 12b shows the distribution of the number of radios changed after each traffic variation. It can be observed that MVCRA-R and MVCRA make approximately the maximum allowed number of changes (about 11.5 on the average both in the Increase and in the Swap cases), while FCRA changes 20 radios on the average, both in the Increase and in the Swap cases.

We conducted ns-3 simulations to evaluate the throughput achieved in the configurations computed by each of the algorithms after each traffic variation. The distribution of the average throughput over the whole simulation time is reported in Fig. 12c for both UDP and TCP traffic. The results show that MVCRA-R enables a throughput increase with respect to both the other algorithms and the strategy of leaving the channels unchanged. Indeed, for UDP traffic, MVCRA-R achieves, on the average, a throughput increase of 28 percent in the Increase case and of 24 percent in the Swap case, while FCRA achieves, on the average, a throughput increase of 14 percent (Increase) and of 21 percent (Swap). MVCRA, instead, achieves, on the average, a throughput increase of 10 percent (Increase) and of 14 percent (Swap). For TCP traffic, MVCRA-R achieves, on the average, a throughput increase of 12 percent in the Increase case and of 23 percent in the Swap case, while FCRA achieves, on the average, a throughput increase of 8 percent (Increase) and of 17 percent (Swap). MVCRA, instead, achieves, on the average, a throughput increase of 2 percent (Increase) and of 10 percent (Swap).

5.1.4 Repeated Variations in the Traffic Demands

The aim of this set of simulations is to evaluate the gain achieved by different channel assignment algorithms with respect to the strategy of leaving the channel assignment unchanged when repeated variations in the traffic demands occurs. The first variation is that described in the previous section under the Increase strategy. The second variation consists in scaling the amount of traffic of each demand by a factor derived from a uniform distribution $U(0.5, 1)$, the third from a distribution $U(0.75, 1.75)$, the fourth from a distribution $U(0.5, 1.5)$. For every variation, each algorithm is given the configuration it computed at the previous traffic variation. $One + MVCRA\text{-}R$ denotes the results achieved by MVCRA-R when the initial configuration is such that each node uses just one radio set to a common channel. Figs. 13a and 13c show the maximum total utilization and average throughput normalized to those achieved with the initial channel assignment



(a) Normalized max total utilization (b) Number of radios changed (c) Normalized UDP throughput

Fig. 13. Repeated traffic variation.

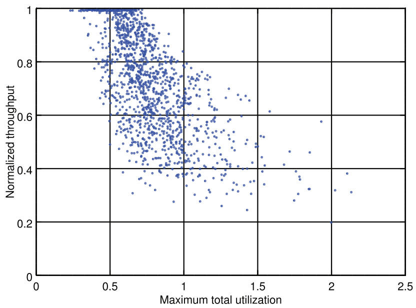


Fig. 14. Correlation (UDP traffic).

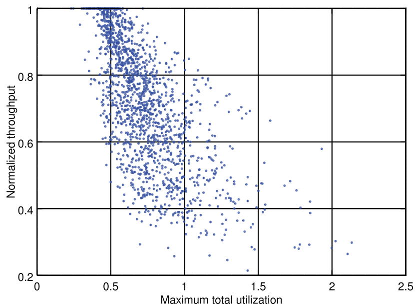


Fig. 15. Correlation (TCP traffic).

computed by FCRA. It can be observed that MVCRA-R outperforms the other algorithms after all the traffic variations. Also, the performance of $One + MVCRA\text{-}R$ rapidly becomes comparable to the best one, thus showing that MVCRA-R is able to adapt to the current configuration and rapidly recover even from very poor configurations. Fig. 13b shows the number of radios changed after each variation. We note that MVCRA-R, when starting from a configuration where a single channel is used, does not meet the constraint on the maximum allowed number of radio changes. This result shows that MVCRA-R, when given a poor network configuration, prefers to rapidly decrease the maximum total utilization, even though that requires to change a higher number of radios. However, it can be observed that after the subsequent traffic variations $One + MVCRA\text{-}R$ tends to change a number of radios comparable to MVCRA-R and MVCRA.

5.1.5 Correlation between Maximum Total Utilization and Throughput

In this section, we consider all the simulations that we performed (whose results have been shown in the previous sections) and, for each simulation, we relate the maximum total utilization with the average throughput normalized to the total amount of all the traffic demands. The results are shown in Fig. 14 (UDP traffic) and Fig. 15 (TCP traffic). It can be observed that, as the maximum total utilization increases, the achieved throughput tends to be a smaller fraction of the traffic demands. To measure the statistical dependence between these two variables, we computed the Spearman rank correlation coefficient [24]. Such coefficient ranges from -1 to 1 and indicates how well the relationship between two variables can be described using a monotonic function. A value of 1 (-1) denotes a perfect monotone increasing (decreasing) relationship. A value between 0.5 and 1 (-0.5 and -1) denotes a strong positive (negative) correlation.

In our analysis, we got a value of -0.785 for UDP traffic and -0.755 for TCP traffic, thus indicating the strong negative correlation between the maximum total utilization and the achieved throughput (normalized to the total traffic load). Such result substantiates our choice to consider the minimization of the maximum total utilization as the objective of our channel reassignment algorithm. Also, this analysis confirms that (2) and (3) can be used to provide reference values for the maximum total utilization that ensures that the network is actually able to carry (a high fraction of) the traffic load. Indeed, the RHS of (2) yields 0.53 with a packet payload size of $1{,}400B$ (the same value used in our simulations). Despite the network links use distinct transmission rates and the collision domains are such that two links may be active simultaneously, it turns out that, for all the simulations where the maximum total utilization is below 0.53, in the 84.5 percent of the cases the achieved throughput is over 95 percent of the traffic load and only in the 3 percent of the cases the achieved throughput is less than the 70 percent of the traffic load. Also, if we only consider the simulations where the maximum total utilization is between 0.53 and 0.70, it turns out that only in the 32.3 percent of the cases the achieved throughput is over 95 percent of the traffic load. A similar trend is observed for the simulations with TCP traffic.

5.1.6 Simulation with Real Traffic Traces

We performed a simulation study where the traffic injected into the network is based on real traffic traces. We considered six traffic traces collected at the gateway router of the wireless network at the University of California, San Diego (UCSD) Computer Science building [25]. Each of such traces records the traffic collected in 1 hour. For each trace, we only considered TCP packets and classified each of them as *upstream* or *downstream*. To this end, we identified all the TCP SYN segments and recorded the corresponding 4-tuple (IP source address, source port, IP destination address, destination port). Then, all the TCP packets matching a 4-tuple have been marked as upstream (TCP connections have been likely opened by the hosts of the wireless network), while TCP packets having source and destination IP addresses and ports swapped with respect to a 4-tuple have been marked as downstream.

We considered topology C (Table 1) and selected two nodes as gateways. Three other mesh nodes act as *aggregation* nodes. Each pair of aggregation node and gateway is associated with a traffic trace. In particular, upstream packets are sent from an aggregation node to a gateway, while downstream packets are sent from a gateway to an aggregation node. In such a way, it is as though the WMN was used as access network by the hosts of the UCSD wireless network. We use the term traffic demand to generically refer to the upstream or downstream flow deriving from a traffic trace.

We considered time slots of duration 10 seconds and computed the average traffic load (deriving from all the six traffic traces) over each time slot. Then, consecutive time slots with similar traffic load are aggregated into time intervals. The average traffic load in each time slot and in each of the 25 time intervals is shown in Fig. 16. The initial

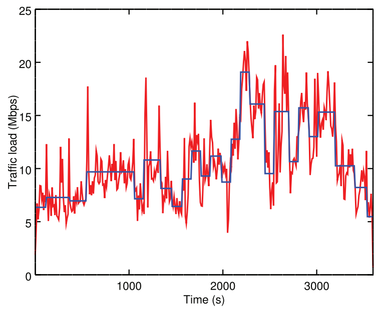


Fig. 16. Traffic traces total load.

flow rates are computed by routing the average load of each traffic demand over the first time interval along the three shortest paths between the corresponding pair of aggregation node and gateway, while the initial channel assignment is computed by FCRA. The initial channel assignment is kept for the duration of the first time interval. At the beginning of every time interval, the maximum total utilization resulting from the channel assignment used in the previous time interval and the average load of each traffic demand over the next time interval is computed. If such maximum total utilization exceeds a given threshold, a new channel assignment is computed. The threshold we use in our experiment is 0.4, i.e., the value obtained by substituting the average packet size resulting from the traffic traces we considered ($850B$) into the RHS of (2). We implemented an ns-3 module to generate traffic according to a given trace. We used static routing in our simulation to avoid the transient throughput loss due to the time required by nodes to discover new links established on new channels (see Section 5.2). We did so because such throughput loss is dependent on the particular routing protocol used and we want to evaluate the throughput achieved in a steady state.

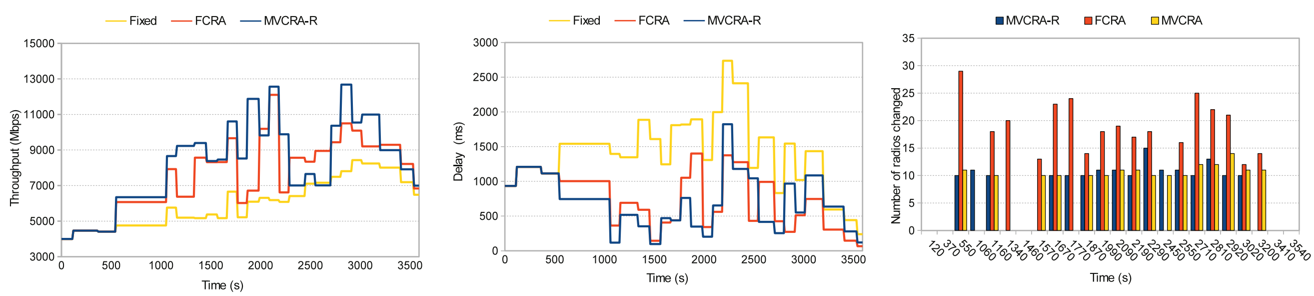
The results of the experiment that we conducted are shown in Fig. 17. The average throughput achieved by each channel assignment algorithm (except MVCRA, which is omitted for clarity) is shown in Fig. 17a. Here, "Fixed" refers to the strategy of leaving the channel assignment unchanged. It can be observed that the fixed strategy achieves the lowest average throughput in every time interval, while MVCRA-R achieves the highest average throughput in almost all the time intervals. The highest average throughput over all the simulation duration is achieved by MVCRA-R (8,163 Mbps), which attains a 9 percent throughput increase with respect to FCRA, 13 percent with respect to MVCRA, and 35 percent with respect to the fixed strategy. The average delay experienced by the packets sent in each time interval is shown in Fig. 17b. It can be observed that the fixed strategy leads to the highest average delay, while MVCRA-R enables a 50 percent reduction, FCRA a 46 percent reduction, and MVCRA a 40 percent reduction.

Fig. 17c shows the number of radios changed by each algorithm at the beginning of each time interval. MVCRA-R and MVCRA change 10.81 radios on the average, thus roughly satisfying the constraint on the maximum allowed number of radio changes and requiring far less radio changes than FCRA (19 on the average). Finally, as far as the maximum total utilization is concerned, it turns out that MVCRA-R achieves the minimum average value (0.43), followed by MVCRA (0.46), FCRA (0.48), and the fixed strategy (0.65).

The results of the simulation conducted with real traffic traces showed that MVCRA-R outperforms FCRA (higher throughput and far less radio changes required) and enables a considerable gain in throughput with respect to the leaving the channel assignment unchanged.

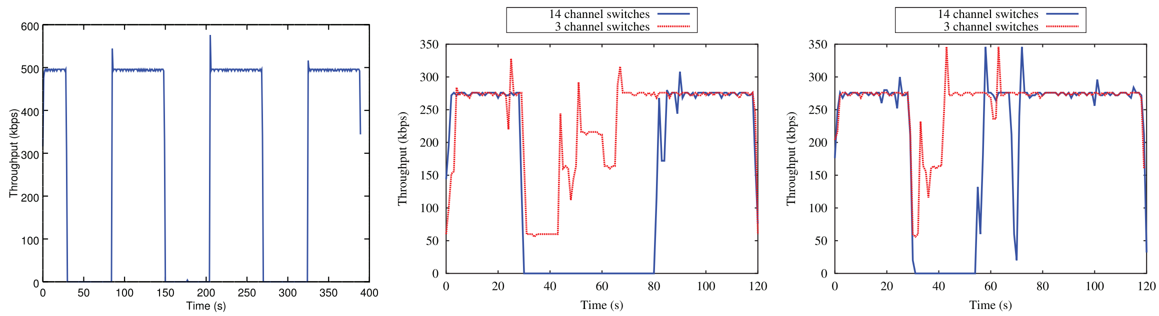
5.2 Experimental Results

We performed experiments using the ORBIT wireless testbed [2] to gain some insight into the effects of switching channels on the network performance. Each node of the testbed has two IEEE 802.11 interfaces and runs Linux (kernel version 2.6.35). We selected optimized link state routing (OLSR), standardized in RFC 3626 [26], as the routing protocol, since there exists an open-source implementation [27] that is deployed on community WMNs with hundreds of nodes and is considered to be solid and stable. We first present the results of a simple experiment involving two nodes. The sender generates TCP traffic at the constant rate of 500 kbps destined to the receiver. The sender has two radios set on channels 36 and 44 (802.11a), while the receiver has a radio set on channel 36 (802.11a). A link is thus established on channel 36. After 30 seconds, the radio on the receiver switches from channels 36 to 44. Then, every 2 minutes we keep switching the radio on the receiver between channels 36 and 44. The average throughput over time slots of 1 second is shown in Fig. 18a. We can notice that, as soon as the first channel switch takes place, there is no connectivity between the sender and the receiver for about 55 seconds. Such a delay can be explained as follows: We recall that OLSR nodes periodically broadcast HELLO messages that are used to determine the quality of the link to a neighbor. To avoid



(a) Average throughput (b) End-to-end delay (c) Number of radios changed

Fig. 17. Simulation with real traffic traces.



(a) Simple experiment (b) Plain OLSR (c) OLSR with link invalidation

Fig. 18. Throughput measured in the experiments in the ORBIT testbed.

fluctuations due to transient noise/interference, RFC 3626 introduces the hysteresis strategy and two thresholds: An established link is no longer used when its quality drops below the lower threshold, while a new link is considered established when its quality exceeds the higher threshold. Hence, a number of HELLO intervals must elapse before the OLSR daemon on the sender considers the link on channel 36 as lost and the new link on channel 44 as established. Until that happens, the sender keeps transmitting using the radio set on channel 36 and thus the packets are not received, as the receiver switched its radio to channel 44. Things are also made worse by TCP, which keeps increasing the retransmission timeout during the absence of connectivity. Thus, when the network connectivity is restored, the sender TCP entity may be waiting for the retransmission timer to expire and thus delays the retransmission of the queued segments until after the timer expires. Indeed, repeating the experiment with UDP traffic showed that the throughput is null for about 42 seconds.

The results shown in Fig. 18a are obtained by using the default value for the HELLO interval (2 seconds). Clearly, all the protocol parameters can be tweaked to reduce the time taken by the routing protocol to switch to using the new link. For instance, we repeated the previous experiment by using an HELLO interval of 1 second and obtained that the connectivity is lost for 28 seconds instead of 55. However, reducing the HELLO interval increases the overhead. Also, the hysteresis thresholds might be changed, but considering a link as established or lost after a few successful/unsuccessful HELLO message transmissions might lead to routing instability. Indeed, an active link may suffer bursty transmission failures due to collisions or transient noise/interference.

We now analyze the effects of multiple channel switches in a bigger network. We used 13 nodes in the ORBIT testbed, two of which are used as senders and other two are used as receivers. Each sender generates two traffic flows destined to each of the receivers. One of the nodes acts as channel assignment server and communicates (via a TCP connection) to each of the other nodes the new channels their radios must be set to. The actual channel switching is triggered by the reception of a UDP message that is sent by the channel assignment server 30 seconds after the start of the experiment and rebroadcast by every node receiving it (before switching channels). In the attempt to speed up the recovery from the breakages caused by switching channels, we enabled the use of the link quality as link metric instead of the simple hop count provided by RFC 3626.[1] Indeed, using the hop count metric, the current path is only replaced when one of its links is marked as lost, while, using the link quality, a path with a better quality than the current one can be preferred even before a link of the current path is marked as lost.

We conducted two experiments. In one case, channels are reassigned by FCRA, which results in 14 channel switches. In the other case, channels are reassigned by MVCRA-R, which results in three channel switches. When TCP is used (Fig. 18b), we can observe that, in the case of 14 channel switches, the time required to restore the steady-state throughput is about 55 seconds, i.e., what we measured in the simple experiment. For most of such time interval, the network throughput is null. The reason is that the (approximately) simultaneous switching of a high number of radios breaks the connectivity between many nodes (as illustrated by the simple experiment). In our case, there is no path between the senders and the receivers made of nodes that keep their connectivity during the channel switching, and hence the throughput is null. In the experiment where only three radios are changed, we can observe that the throughput drops to 60 kbps right after the channel switching. That happens because the path between a single sender-receiver pair is not affected by the channel switches. Also, the small number of radio changes allows to find alternate paths for the other sender-receiver pairs that do not include links affected by the channel switches. Those alternate paths are preferred, and hence used, by the routing protocol as soon as the quality of the current paths drops below their quality. Actually, that happens before the links using the new channels are considered established. The result is that restoring all the sender-receiver pairs takes about 35 seconds and thus is quicker than in the previous case. When UDP is used, a similar behavior is obtained, but the steady-state throughput is restored more quickly (42 and 30 seconds in case of 14 and 3 channel switches, respectively), thus confirming the additional delay introduced by TCP mechanisms.

The previous experiments show that identifying the links that are established/lost following a radio channel switch by means of the usual procedures based on the link quality is not efficient. We believe that a cross-layer exchange of information between the routing and MAC layers may help

1. A second version of OLSR is being worked on within the IETF that enables the use of metrics other than the hop count.

speed up the process of switching to the newly established link. Though the design of a cross-layer solution is left for future work, here we present the results of a preliminary investigation. We repeated the previous experiments, with the difference that, just before switching channel on a radio, a node informs both its neighbors (through an HELLO message) and all the other nodes (through a TC message) that the links using that radio are lost and then invalidates such links. In case of TCP traffic (Fig. 18c), the steady-state throughput is restored after 29 and 13 seconds in case of 14 and 3 channel switches, respectively. Thus, the proposed mechanism helps to reduce the time needed to restore the network throughput. Nevertheless, we observe that limiting the number of radios switching channel is still beneficial, because more paths are unaffected and can be immediately used by the routing protocol to replace those affected by the link breakages.

6 Conclusions

In this paper, we presented the MVCRA-R algorithm, which takes the current channel assignment and the new set of flow rates into account and attempts to minimize the maximum total utilization over all the collision domains while constraining the number of radios that can be assigned a new channel. With respect to MVCRA, MVCRA-R leverages the possibility to adjust the link transmission rates and presents some enhancements such as an improved definition of the link priorities. We performed extensive simulation studies that confirmed that MVCRA-R roughly meets the constraint on the maximum allowed number of radio changes and outperforms both MVCRA and a channel assignment algorithm such as FCRA in terms of maximum total utilization and network throughput. The simulation studies also confirmed the strong correlation between the maximum total utilization and the throughput, thus supporting our choice for the objective function of MVCRA-R. Also, we conducted experiments in a real wireless testbed to evaluate how switching channels affects the network performance. We believe that investigating measures to limit such impact constitutes an interesting subject for future work.

Acknowledgments

This research was supported by grant YR2009-7003 from Stiftelsen för interntionalisering av högre utbildning och forskning (STINT).

References

[1] S. Avallone, I.F. Akyildiz, and G. Ventre, “A Channel and Rate Assignment Algorithm and a Layer-2.5 Forwarding Paradigm for Multi-Radio Wireless Mesh Networks,” *IEEE/ACM Trans. Networking,* vol. 17, no. 1, pp. 267-280, Feb. 2009.

[2] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, “Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols,” *Proc. IEEE Wireless Comm. Networking Conf. (WCNC),* vol. 3, pp. 1664-1669, Mar. 2005.

[3] J. Tang, G. Xue, and W. Zhang, “Interference-Aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks,” *Proc. ACM MobiHoc,* pp. 68-77, 2005.

[4] K.N. Ramachandran, E.M. Belding, K.C. Almeroth, and M.M. Buddhikot, “Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks,” *Proc. IEEE INFOCOM,* 2006.

[5] K. Xing, X. Cheng, L. Ma, and Q. Liang, “Superimposed Code Based Channel Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks,” *Proc. ACM MobiCom,* pp. 15-26, 2007.

[6] H. Skalli, S. Ghosh, S. Das, L. Lenzini, and M. Conti, “Channel Assignment Strategies for Multiradio Wireless Mesh Networks: Issues and Solutions,” *IEEE Comm. Magazine,* vol. 45, no. 11, pp. 86-95, Nov. 2007.

[7] A. Subramanian, H. Gupta, S.R. Das, and J. Cao, “Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks,” *IEEE Trans. Mobile Computing,* vol. 7, no. 12, pp. 1459-1473, Dec. 2008.

[8] A. Raniwala and T. Chiueh, “Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network,” *Proc. IEEE INFOCOM,* vol. 3, pp. 2223-2234, 2005.

[9] A. Dhananjay, H. Zhang, J. Li, and L. Subramanian, “Practical, Distributed Channel Assignment and Routing in Dual-Radio Mesh Networks,” *Proc. ACM SIGCOMM,* pp. 99-110, Aug. 2009.

[10] A. Raniwala, K. Gopalan, and T. Chiueh, “Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks,” *ACM Mobile Computing and Comm. Rev.,* vol. 8, no. 2, pp. 50-65, Apr. 2004.

[11] M. Alicherry, R. Bhatia, and E. Li, “Joint Channel Assignment and Routing for Throughput Optimization in Multiradio Wireless Mesh Networks,” *IEEE J. Selected Areas in Comm.,* vol. 24, no. 11, pp. 1960-1971, Nov. 2006.

[12] M. Kodialam and T. Nandagopal, “Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks,” *Proc. ACM MobiCom,* pp. 73-87, 2005.

[13] X. Lin and S. Rasool, “A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad-Hoc Wireless Networks,” *Proc. IEEE INFOCOM,* pp. 1118-1126, 2007.

[14] A.M. Rad and V. Wong, “Joint Logical Topology Design, Interface Assignment, Channel Allocation, and Routing for Multi-Channel Wireless Mesh Networks,” *IEEE Trans. Wireless Comm.,* vol. 6, no. 12, pp. 4432-4440, Dec. 2007.

[15] J. Crichigno, M.-Y. Wu, and W. Shu, “Protocols and Architectures for Channel Assignment in Wireless Mesh Networks,” *Ad Hoc Networks,* vol. 6, no. 7, pp. 1051-1077, 2008.

[16] Y. Wu, M. Keally, G. Zhou, and W. Mao, “Traffic-Aware Channel Assignment in Wireless Sensor Networks,” *Proc. Int’l Conf. Wireless Algorithms, Systems, and Applications (WASA),* pp. 479-488, 2009.

[17] G. Zhou, L. Lu, S. Krishnamurthy, M. Keally, and Z. Ren, “SAS: Self-Adaptive Spectrum Management for Wireless Sensor Networks,” *Proc. 18th Int’l Conf. Computer Comm. and Networks,* pp. 1-6, Aug. 2009.

[18] G. Zhou, T. He, J. Stankovic, and T. Abdelzaher, “RID: Radio Interference Detection in Wireless Sensor Networks,” *Proc. IEEE INFOCOM,* vol. 2, pp. 891-901, Mar. 2005.

[19] A. Franklin, A. Balachandran, C. Murthy, and M. Marina, “Demand Based State Aware Channel Reconfiguration Algorithm for Multi-Channel Multi-Radio Wireless Mesh Networks,” *Proc. IEEE INFOCOM,* pp. 1-6, 2010.

[20] J. Rezgui, A. Hafid, R. Ben Ali, and M. Gendreau, “Meta-Heuristics for Channel (Re-)Assignment Problem for Multi-Radio Wireless Mesh Networks,” *Proc. INFORMS Telecom Conf.,* May 2010.

[21] S. Avallone, F. D’Elia, and G. Ventre, “A Traffic-Aware Channel Re-Assignment Algorithm for Wireless Mesh Networks,” *Proc. European Wireless Conf.,* pp. 683-688, Apr. 2010.

[22] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, “Impact of Interference on Multi-hop Wireless Network Performance,” *Proc. ACM MobiCom,* pp. 66-80, 2003.

[23] D. Eppstein, “Finding the k Shortest Paths,” *SIAM J. Computing,* vol. 28, no. 2, pp. 652-673, Apr. 1999.

[24] C. Spearman, “The Proof and Measurement of Association between Two Things,” *Am. J. Psychology,* vol. 15, no. 1, pp. 72-101, Jan. 1904.

[25] Y.-C. Cheng, “CRAWDAD Data Set UCSD/CSE (v. 2008-08-25),” http://www.crawdad.org/download/ucsd/cse/wireless, 2013.

[26] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” IETF RFC 3626, Oct. 2003.

[27] “OLSR Daemon,” http://www.olsr.org, 2013.

Stefano Avallone received the MS degree in telecommunications engineering in 2001 and the PhD degree in computer networks in 2005 from the University of Napoli "Federico II." He is currently an assistant professor in the Department of Computer Engineering at the University of Napoli. His research interests include computer networks, traffic engineering, QoS routing, and wireless mesh networks. He was a visiting researcher at the Delft University of Technology (2003-2004) and at the Georgia Institute of Technology (2005). He is on the editorial board of Elsevier's *Ad Hoc Networks*. He is a member of the IEEE.

Giovanni Di Stasi received the Laurea degree in computer engineering from the University of Napoli Federico II in 2007. He is currently working toward the PhD degree at the University of Napoli Federico II. He was a visiting researcher at INRIA Sophia Antipolis, France (2009), and at the Karlstad University, Sweden (2010). His current research interests include experimental research infrastructures and testbeds, routing and channel assignment algorithms for wireless mesh networks, and peer-to-peer traffic optimization.

Andreas Kassler received the MSc degree in mathematics/computer science from Augsburg University, Germany, in 1995 and the PhD degree in computer science from the University of Ulm, Germany, in 2002. He is currently a full professor in the Department of Computer Science at Karlstad University in Sweden, where he teaches wireless networking and advanced topics in computer networking. His main research interests include wireless meshed networks, ad hoc networks, future Internet, multimedia networking, QoS, and P2P systems. He is a member of the IEEE.