Computer Networks 53 (2009) 2502–2516

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Congestion-aware channel assignment for multi-channel wireless mesh networks ☆

A. Hamed Mohsenian Rad, Vincent W.S. Wong *

Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada

ARTICLE INFO

Article history:
Received 10 July 2008
Received in revised form 21 January 2009
Accepted 11 May 2009
Available online 23 May 2009

Responsible Editor: G. Morabito

Keywords:
Multi-channel wireless mesh networks
Distributed algorithm
Channel assignment
Congestion control
Cross-layer design
Frequency overlapping

ABSTRACT

In this paper, we propose a distributed congestion-aware channel assignment (DCACA) algorithm for multi-channel wireless mesh networks (MC–WMNs). The frequency channels are assigned according to the congestion measures which indicate the congestion status at each link. Depending on the selected congestion measure (e.g., queueing delay, packet loss probability, and differential backlog), various design objectives can be achieved. Our proposed distributed algorithm is simple to implement as it only requires each node to perform a local search. Unlike most of the previous channel assignment schemes, our proposed algorithm assigns not only the non-overlapped (i.e., orthogonal) frequency channels, but also the partially-overlapped channels. In this regard, we introduce the *channel overlapping* and *mutual interference* matrices which model the frequency overlapping among different channels. Simulation results show that in the presence of elastic traffic (e.g., TCP Vegas or TCP Reno) sources, our proposed DCACA algorithm increases the aggregate throughput and also decreases the average packet round-trip compared with the previously proposed *Load-Aware* channel assignment algorithm. Furthermore, in a congested IEEE 802.11b network setting, compared with the use of three non-overlapped channels, the aggregate network throughput can further be increased by 25% and the average round-trip time can be reduced by more than one half when all the 11 partially-overlapped channels are used.

1. Introduction

Wireless mesh networks (WMNs) have recently received an increasing attention to provide ubiquitous and inexpensive last-mile Internet access. A WMN consists of a number of stationary wireless mesh routers, forming a wireless backbone. These routers serve as access points for various wireless mobile devices. Some of the routers also act as gateways to the Internet via high-speed wired links. Mobile devices first transfer data to their associated router, and the data is then transferred to the Internet via the intermediate routers in a multi-hop manner [1,2].

The aggregate capacity and the performance of WMNs can be increased by the use of multiple channels [3]. In this scenario, each wireless mesh router is equipped with multiple network interface cards (NICs). Each NIC operates on a distinct frequency channel in the IEEE 802.11a/b/g bands. Two neighboring routers can communicate with each other if each one has an NIC operating on the same frequency channel. Within the IEEE 802.11 frequency bands, the number of available channels is limited. For example, the IEEE 802.11b/g standards have 11 channels, of which three channels are non-overlapped. The number of operating channels in the IEEE 802.11a standard is 79, of which 12 channels are non-overlapped. These imply that some logical links may be assigned to the same channel. Interference will occur if these links are close to each other.

☆ This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant Number 261604-07. This paper was presented in part at the *IEEE ICC*, Glasgow, Scotland, UK, June 2007.

* Corresponding author. Tel.: +1 604 827 5135.
E-mail addresses: hamed@ece.ubc.ca (A.H. Mohsenian Rad), vincentw@ece.ubc.ca (V.W.S. Wong).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.05.006

Interference among neighboring links can reduce their effective data rate and potentially cause network congestion. For applications which use TCP (transmission control protocol) in the transport-layer, if the links become congested, there will be a reduction of the aggregate throughput as well as their quality-of-service. Thus, efficient channel assignment is crucial to reduce interference among neighboring transmissions.

There exists a wide range of proposed channel assignment algorithms for multi-channel wireless mesh networks (MC–WMNs) in the literature. One approach is to formulate channel assignment problem as an optimization problem [4–18]. Das et al. [5] proposed an optimization-based algorithm that maximizes the number of logical links that can be active simultaneously, subject to interference constraints. Chen et al. [6] devised a channel assignment strategy which assigns the channels in order to balance the traffic load between different channels. The algorithms in [5–8] are *static* and assign the channels *permanently*. There also exist some *dynamic* algorithms which update the assigned channels either in a *short-term* basis (e.g., packet-by-packet [9–11]) or a *long-term* basis (e.g., every several minutes or hours [12,13,15,16,14]). Short-term channel updates require fast channel switching which can be a challenge in the existing commercial IEEE 802.11 interfaces with a switching latency in the order of 100 ms [19,20]. Another challenge is the required fast coordination to ensure that the sending and receiving routers use the same channel. On the other hand, long-term interval channel updates do not require fast switching and coordination. They can also use the existing IEEE 802.11 commodities. Raniwala and Chiueh [15] proposed a long-term dynamic channel assignment algorithm called *Load-Aware* algorithm. By monitoring the amount of traffic being transmitted over each frequency channel, wireless mesh routers assign their NICs with those channels which have minimum usage within their neighborhood. In [12], Alicherry et al. proposed an interference-free scheduler which aims in maximizing the bandwidth allocated to each router subject to the constraint that the allocated bandwidth is in proportion to its aggregate traffic demand. Kodialam and Nandagopal [13] also proposed an algorithm that maximizes the network throughput subject to the minimum rate requirements for each flow. It has been also recently shown in two independent works in [21,22] that using partially-overlapped frequency channels can further improve network performance. Last but not least, the study of channel assignment when smart directed antennas are used is presented in [23]. Non-cooperative channel assignment is also studied in [24].

In summary, most of the previous channel assignment algorithms mentioned above have one or more of the following performance bottlenecks: First, many of these algorithms are *centralized*. They require strong coordination and result in high computational complexity and significant signalling overhead. Second, they only take into account the orthogonal (i.e., non-overlapped) frequency channels, but not the partially overlapped channels. Thus, the frequency spectrum is not utilized efficiently. Third, some of these algorithms are static. They cannot adapt to the time-varying features of the networks such as the variable traffic demands. Fourth, most of the previous algorithms are based on various *heuristic* design. This may be due to the lack of accurate capacity models in terms of the channel assignment variables. Finally, none of the algorithms mentioned above take congestion information into account.

In this paper, we propose the distributed congestion-aware channel assignment (DCACA) algorithm which overcomes the above performance bottlenecks in all five aspects. Our proposed algorithm is distributed and is executed by each wireless mesh router in an asynchronous manner. In this regard, each node only needs to perform a simple local search to adequately assign the frequency channels to a subset of logical links. Moreover, our proposed algorithm is able to assign not only the non-overlapped channels, but also all the available partially-overlapped channels with arbitrary overlapping portions. In this regard, we propose two key matrices, called *channel overlapping matrix* and *mutual interference matrix*, that are able to mathematically model the frequency overlapping among the channels. Our proposed algorithm is a long-term dynamic channel assignment algorithm. It assigns the frequency channels based on the most recent congestion information measured across the network. Unlike other distributed channel assignment algorithms which suggest *selfish* actions by each wireless mesh router, our proposed algorithm is based on *cooperation* among the routers. This is indeed necessary for achieving the optimal network performance in a distributed fashion. Finally, our algorithm is designed to solve a mathematically formulated *congestion-aware* channel assignment problem in the general form of maximizing a weighted summation of link capacities. Depending on the selected weighting parameters, solving our formulated problem results in achieving some well-known resource allocation design objectives for both fixed and elastic traffic patterns. In this regard, we derive a closed-form capacity model for each logical link in terms of our defined channel assignment variables as well as the channel overlapping and mutual interference matrices. Simulation results show that if TCP Vegas is used, then our proposed DCACA algorithm increases the aggregate throughput by 11.5% and decreases the average packet round-trip time by 35.3% compared to the *Load-Aware* algorithm [15]. On the other hand, if TCP Reno is used, then DCACA algorithm increases the aggregate throughput by 9.8% and decreases the average packet round-trip time by 28.7% compared to the *Load-Aware* algorithm [15]. Furthermore, in a congested IEEE 802.11b wireless network setting, compared with the use of 3 non-overlapped frequency channels, the aggregate throughput can further be increased by 25% and the average round-trip time can be reduced by more than one half when all the 11 partially-overlapped channels are used.

The rest of this paper is organized as follows. The problem formulation is described in Section 2. Our proposed link capacity models are developed in Section 3. In Section 4, we propose the DCACA algorithm and provide the proof of its convergence. Performance assessments and comparison studies are presented in Section 5. Concluding remarks are given in Section 6. A list of the key notations that we used in this paper is shown in Table 1.

Table 1
Index of key notations

$\mathscr{N}, N$	Set of nodes (i.e., wireless mesh routers) and the number of nodes, respectively
$\mathscr{L}, L$	Set of links and the number of links, respectively
$\mathscr{C}, C$	Set of channels and the number of channels, respectively
$\mathscr{I}_n, I_n$	Set of NICs in node n and the number of NICs in node n, respectively
I	Maximum number of NICs among all nodes
$\mathscr{L}_n^{\text{in}}, \mathscr{L}_n^{\text{out}}$	Set of all incoming and outgoing links of node n, respectively
$\mathscr{L}_n, \mathscr{L}_{n,i}$	Set of links of node n and set of links using NIC i on node n, respectively
$\mathbf{x}_l$	Channel assignment vector corresponding to link l
$\mathbf{x}$	Channel assignment vector corresponding to all links
$\mathscr{X}$	Set of all feasible channel assignment vectors
q_l, λ_l	Congestion measure of link l and the persistent probability of link l, respectively
$\boldsymbol{\lambda}, \mathbf{q}$	Congestion measure of all links and the persistent probability of all links, respectively
c_l	Capacity of link l
T_G	Time interval at which problem (CACA) needs to be solved
T_L	Time interval at which problem (LOCAL-CACA) is being solved
$\gamma_l, \gamma_{\min}$	SINR for link l and minimum required SINR, respectively
T_S	Symbol period
K	A constant which depends on the modulation scheme
F_u	Power spectral density function of the band-pass filter for channel u
$\mathbf{W}$	Channel overlapping matrix
w_{uv}	Entry in the uth row and the vth column of channel overlapping matrix $\mathbf{W}$
$\mathbf{M}_{lk}$	Mutual interference matrix corresponding to links l and k
g_{lk}	Path loss from the transmitter node of link l to the transmitter node of link k
e_{lk}	Euclidian distance between the transmitter node of link l and the transmitter node of link k
κ	Path loss exponent
p_l	Transmission power of the transmitter node of link l
η_l	Noise power at the receiver node of link l
δ	Roll-off factor
α	A constant which depends on the antenna gains and signal wavelengths
$\mathscr{O}_l, O_l$	Opponent set of link l and the number of opponent links of link l, respectively
$\circ$	Hadamard product
s_l	The node which is responsible for channel assignment of link l
$\mathbf{x}_{-n}$	Channel assignment vectors corresponding to all links other than links of node n
$L^{\max}$	Maximum links connected to any node in the network
r_{sd}	Data rate for the end-to-end traffic from source node s to destination node d
$\mathscr{T}$	Set of all time slots
$\mathscr{T}_G$	Set of all time slots at which problem (CACA) needs to be solved
$\mathscr{T}_{L,n}$	Set of all time slots at which problem (LOCAL-CACA) is being solved by node n
$\psi(t)$	The objective function of problem (CACA) at time slot t

2. Problem formulation

Consider an MC–WMN with $\mathscr{N}$ as the set of wireless nodes (i.e., wireless mesh routers) and $\mathscr{L}$ as the set of uni-directional logical links[1]. We define $N = |\mathscr{N}|$ and $L = |\mathscr{L}|$ as the cardinality of set $\mathscr{N}$ and $\mathscr{L}$, respectively. For each node $n \in \mathscr{N}$, let $\mathscr{L}_n^{\text{out}}$ denote the set of all *outgoing* links from node n and $\mathscr{L}_n^{\text{in}}$ denote the set of all *incoming* links to node n. We define $\mathscr{L}_n = \mathscr{L}_n^{\text{out}} \cup \mathscr{L}_n^{\text{in}}$. Thus, $\cup_{n \in \mathscr{N}} \mathscr{L}_n = \mathscr{L}$. Let $\mathscr{C}$ denote the set of all available frequency channels and $C = |\mathscr{C}|$ denote the cardinality of set $\mathscr{C}$. For each node $n \in \mathscr{N}$, we also define $\mathscr{I}_n$ as the set of its NICs. The cardinality of set $\mathscr{I}_n$ is denoted by I_n. We have $I = \max_{n \in \mathscr{N}} I_n$. The logical topology is assumed to be symmetric. That is, if $l \in \mathscr{L}_n^{\text{out}} \cap \mathscr{L}_m^{\text{in}}$ is a logical link in the direction from node n to node m, then there exists another link $k \in \mathscr{L}_n^{\text{in}} \cap \mathscr{L}_m^{\text{out}}$ in the direction from node m to node n. The logical topology is also assumed to be *ripple-effect free* [16,15,17]. Two sample ripple-effect free MC–WMN logical topologies are *Hyacinth* [15] and *TiMesh* [17]. They are shown in Fig. 1a and b, respectively. In a ripple-effect free logical topology, there exists an *exclusive* (i.e., not shared) NIC in *at least* one end of *each* logical link. This limits the *channel dependency* among the links. Hence, assigning a new channel to one link does not trigger a series of channel re-assignments across the network (see Fig. 3 in [15]). Thus, distributed channel assignment is feasible. For each node $n \in \mathscr{N}$ and any of its NICs $i \in \mathscr{I}_n$, we define $\mathscr{L}_{n,i}$ as the set of links that use NIC i in node n. Notice that $\cup_{i \in \mathscr{I}_n} \mathscr{L}_{n,i} = \mathscr{L}_n$. Considering the MC–WMN logical topology in Fig. 1a, $\mathscr{L}_a^{\text{out}} = \{l_1, l_3\}, \mathscr{L}_a^{\text{in}} = \{l_2, l_4\}, \mathscr{L}_a = \{l_1, l_2, l_3, l_4\}$, $\mathscr{L}_{a,1} = \{l_1, l_2\}$, and $\mathscr{L}_{a,2} = \{l_3, l_4\}$. We also have: $\mathscr{L}_c^{\text{out}} = \{l_2, l_5, l_7\}, \mathscr{L}_c^{\text{in}} = \{l_1, l_6, l_8\}, \mathscr{L}_c = \{l_1, l_2, l_5, l_6, l_7, l_8\}$, $\mathscr{L}_{c,1} = \{l_1, l_2\}$, and $\mathscr{L}_{c,2} = \{l_5, l_6, l_7, l_8\}$. On the other hand, in Fig. 1b, $\mathscr{L}_d^{\text{out}} = \{l_4, l_9, l_{11}, l_{14}\}, \mathscr{L}_d^{\text{in}} = \{l_3, l_{10}, l_{12}, l_{13}\}, \mathscr{L}_d = \{l_3, l_4, l_9, l_{10}, l_{11}, l_{12}, l_{13}, l_{14}\}$, $\mathscr{L}_{d,1} = \{l_3, l_4, l_9, l_{10}\}$, and $\mathscr{L}_{d,2} = \{l_{11}, l_{12}, l_{13}, l_{14}\}$.

For each logical link $l \in \mathscr{L}$, we define a $C \times 1$ binary *channel assignment vector* $\mathbf{x}_l$. The ith entry of $\mathbf{x}_l$ is equal to 1 if channel i is assigned to link l; otherwise, it is equal to 0. For example, if $C = 4$ and the third channel is assigned to logical link l, then $\mathbf{x}_l = [0 \quad 0 \quad 1 \quad 0]^T$. Since one frequency channel is assigned to each logical link, one of the entries of $\mathbf{x}_l$ should be equal to 1 and the rest should be 0. This requires that:

$$\mathbf{1}^T \mathbf{x}_l = 1, \quad \forall l \in \mathscr{L}, \tag{1}$$

where $\mathbf{1}$ denotes a $C \times 1$ vector with all entries equal to 1. From (1), it is clear that for any two arbitrary links $l, k \in \mathscr{L}$, if they operate over the same channel, then $\mathbf{x}_l^T \mathbf{x}_k = 1$; otherwise, $\mathbf{x}_l^T \mathbf{x}_k = 0$. On the other hand, since all links which share the same NIC need to use the same frequency channel, for each wireless node $n \in \mathscr{N}$ and any of its NICs $i \in \mathscr{I}_n$, we have:

$$\mathbf{x}_l = \mathbf{x}_k, \quad \forall l, k \in \mathscr{L}_{n,i}. \tag{2}$$

For the simplicity of exposition, we stack up the channel assignment vectors corresponding to all links and denote the obtained $LC \times 1$ vector by $\mathbf{x}$. In this paper, we are interested in finding $\mathbf{x}$ to solve the following *global* congestion-aware channel assignment (CACA) problem:

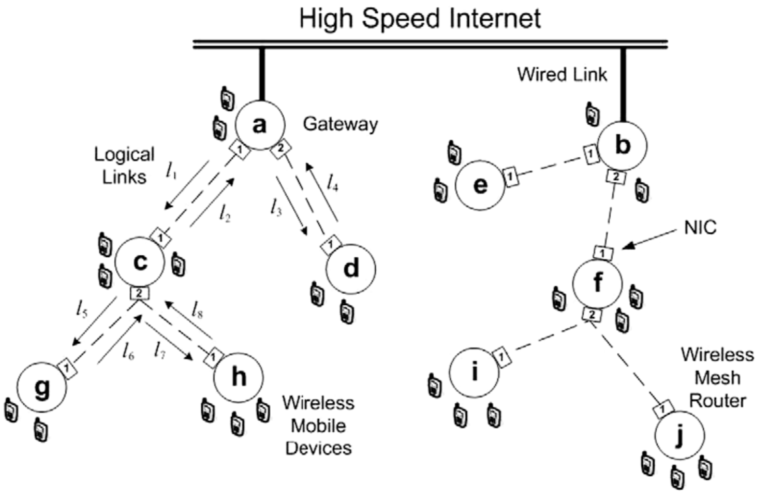
$$\underset{\mathbf{x} \in \mathscr{X}}{\text{maximize}} \sum_{l \in \mathscr{L}} \lambda_l c_l(\mathbf{x}), \tag{CACA}$$

where

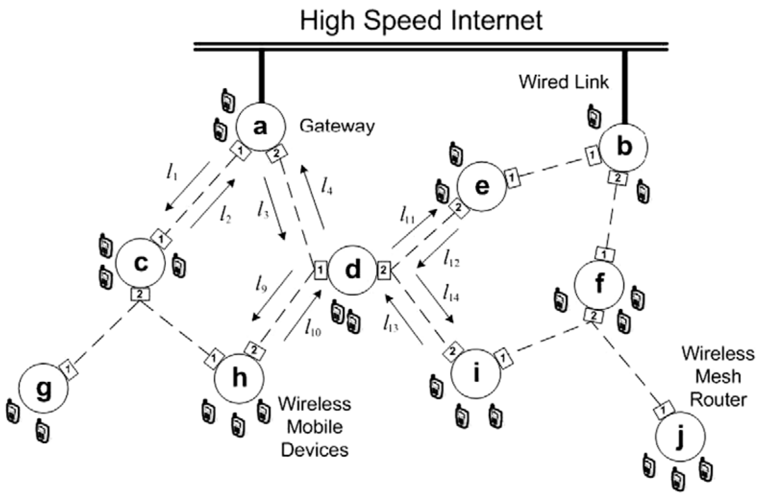
$$\mathscr{X} = \{\mathbf{x} : \ \mathbf{x} \in \{0,1\}^{LC}, \ \mathbf{1}^T \mathbf{x}_l = 1, \ \mathbf{x}_l = \mathbf{x}_k, \ \forall n \in \mathscr{N}, \ i \in \mathscr{I}_n, \ l, k \in \mathscr{L}_{n,i}\}. \tag{3}$$

Here $\{0,1\}^{LC}$ denotes the set of all $LC \times 1$ binary vectors and $\mathscr{X}$ denotes the set of all *feasible* channel assignment vectors.

[1] Here we assume that all logical links in set $\mathscr{L}$ are used for packet transmissions. If there is a logical link that does not belong to any of the routing paths in the network, we simply assume that it does not exist.



(a) *Hyacinth* Logical Topology [15]



(b) *TiMesh* Logical Topology [17]

Fig. 1. Two sample ripple-effect free MC–WMN topologies with 10 wireless mesh routers and 25 wireless mesh clients.

The notations $c_l(\mathbf{x})$ and λ_l denote the capacity and a *congestion measure* of logical link $l \in \mathscr{L}$, respectively. The more the link l is congested, the higher the value of λ_l will be. By solving problem (CACA), the frequency channels are assigned to maximize a weighted summation of link capacities where the congestion measures act as weights. In this regard, the congested logical links are provided with more capacities. Various congestion measures can be considered. For example, following the steps in [22], we can show that the *network utility maximization* (NUM) problem [25–28] in the presence of *elastic* traffic sources can be reduced to solving problem (CACA) if λ_l denotes the *congestion price* on link l. In this case, the congestion prices depend on the transport-layer protocol is being used. For example, the congestion prices are *queueing delay* and *packet loss probability* for TCP Vegas [29] and TCP Reno [30], respectively. On the other hand, we may choose λ_l to be the *differential backlog* corresponding to logical link $l \in \mathscr{L}$. That is, the difference between the queue backlogs in the transmitter and receiver nodes of logical link l. The differential backlog is an indication of *relative congestion*. In this case, solving problem (CACA) results in finding the optimal solution of the *maximum weight matching* (MWM) problem [31,32], which stabilizes the constrained queueing systems and leads to maximum aggregate network throughput. For the simplicity of exposition, we assume that time is divided into equal-length slots $\mathscr{T} = \{0, 1, 2, \ldots\}$. In practice, regardless of the selected congestion measures, we are interested in solving problem (CACA) *periodically*, e.g., every T_G time slots. This is because the congestion measures are usually *time-varying*. Interval parameter T_G can be in the order of several seconds, a couple of minutes, or a few hours depending on the selected congestion measures. Let $\mathscr{T}_G \subset \mathscr{T}$ denote the set of all time slots at which problem (CACA) needs to be solved.

Any two consecutive members of set $\mathcal{T}_G$ should be exactly T_G time units away from each other. For example, $\mathcal{T}_G = \{1, 1+T_G, 1+2T_G, 1+3T_G, \ldots\}$.

3. Link capacity as a function of channel assignment vector

In an MC–WMN, the capacity of a logical link is a function of several parameters including the transmission powers, node positions, and the assigned frequency channels. Since the nodes are stationary, their positions are known in advance. In this paper, we limit our study to the fixed transmission powers. Thus, the link capacity is only a function of the channel assignment vector $\mathbf{x}$. The information-theoretic capacity of the logical link $l \in \mathcal{L}$ can be expressed as [33]:

$$c_l(\mathbf{x}) = \frac{1}{T_S}\log(1 + K\gamma_l(\mathbf{x})), \tag{4}$$

where $\gamma_l(\mathbf{x})$ denotes the *signal to interference and noise ratio* (SINR) of link l, T_S is the symbol period, and K is a constant which depends on the modulation scheme being used. Next, we show that the value of $\gamma_l(\mathbf{x})$ can be determined in the presence of both non-overlapped and partially-overlapped frequency channels.

Assume that u and v are two available channels within the IEEE 802.11 frequency band (i.e., $u, v \in \mathcal{C}$). Let $F_u(\omega)$ and $F_v(\omega)$ denote the transfer functions of the band-pass filters for frequency channels u and v, respectively. The PSD functions can be obtained from the channels' frequency responses. Without loss of generality, we assume the use of *raised cosine filters* [33]. An important parameter to identify the frequency response of this filter is the roll-off factor δ. Fig. 2 shows the response of the IEEE 802.11b channels with δ equals to 1 and 0.25, respectively. We can see that the lower the roll-off factor, the smaller is the *overlapping portion* among the neighboring channels. To model the overlapping among different channels, we define a symmetric $C \times C$ *channel overlapping matrix* $\mathbf{W}$. The entry in the uth row and the vth column of $\mathbf{W}$ is denoted by scalar w_{uv} and is defined as follows:

$$w_{uv} = \frac{\int_{-\infty}^{\infty} F_u(\omega)F_v(\omega)d\omega}{\int_{-\infty}^{\infty} F_u^2(\omega)d\omega}. \tag{5}$$

Let p_l denote the transmission power of the transmitter node of link l. Also, let g_{lk} denote the *path loss* from the transmitter node of link l to the receiver node of link k. Assuming that both links l, k are active, the interference power from link l on link k can be modeled as:

$$\mathbf{x}_l^T\mathbf{W}\mathbf{x}_k g_{lk} p_l = w_{uv} g_{lk} p_l. \tag{6}$$

3.1. All-at-once scheduling

We first consider the *all-at-once scheduling* model. In this model, all the links can be active simultaneously and there is *no* carrier sensing mechanism in the MAC protocol. From (6),

$$\gamma_l(\mathbf{x}) = (g_{ll}p_l) \Big/ \left(\sum_{k\in\mathcal{L}\setminus\{l\}} \mathbf{x}_k^T\mathbf{W}\mathbf{x}_l g_{kl} p_k + \eta_l\right), \quad \forall\, l \in \mathcal{L}, \tag{7}$$

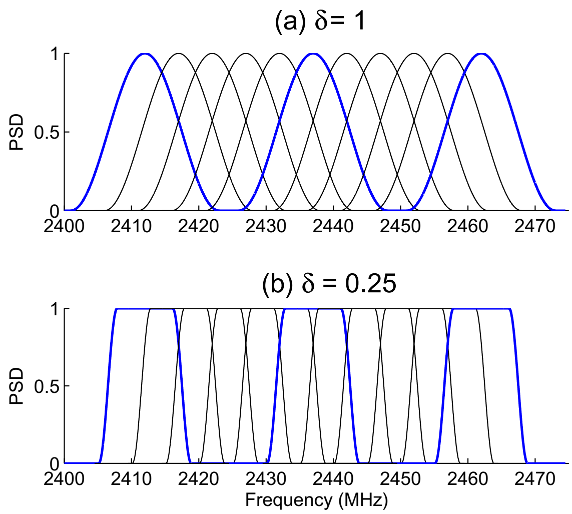


Fig. 2. The available eleven partially-overlapped channels in 802.11b frequency band for roll-off factor $\delta = 1$ and $\delta = 0.25$. The number on each curve indicates the corresponding channel number. Channels 1, 6, and 11 are non-overlapped (orthogonal).

where η_l denotes the thermal noise power at the receiver node of link l. Replacing (7) in (4), we can obtain the capacity model in the all-at-once scheduling scenario as:

$$c_l(\mathbf{x}) = \frac{1}{T_S}\log\left(1 + \frac{Kg_{ll}p_l}{\sum_{k\in\mathcal{L}\setminus\{l\}} \mathbf{x}_k^T\mathbf{W}\mathbf{x}_l g_{kl}p_k + \eta_l}\right), \quad \forall\, l \in \mathcal{L}. \tag{8}$$

From (8), problem (CACA) becomes:

$$\text{maximize}_{\mathbf{x}\in\mathcal{X}} \sum_{l\in\mathcal{L}} \frac{\lambda_l}{T_S} \times \log\left(1 + \frac{Kg_{ll}p_l}{\sum_{k\in\mathcal{L}\setminus\{l\}} \mathbf{x}_k^T\mathbf{W}\mathbf{x}_l g_{kl}p_k + \eta_l}\right). \tag{9}$$

Notice that $1/T_S$ is multiplied to all the terms in the summation. Thus, the value of the symbol period T_S does not affect the solution of the channel assignment problem in (9). On the other hand, in most practical cases, the multiplication factor K is large [34]. For example, in the IS-95 CDMA which is indeed a direct-sequence spread-spectrum system, the processing gain is 128 [34, Chapter 3.4.3, p. 91]. As a result, $\log(1 + K\gamma_l(\mathbf{x})) \approx \log(K\gamma_l(\mathbf{x}))$. This *high SINR regime* approximation is widely used in the networking literature (cf. [35–37]).

In this case, the objective function in problem (9) can be written as:

$$\sum_{l\in\mathcal{L}} \lambda_l \log(Kg_{ll}p_l) - \sum_{l\in\mathcal{L}} \lambda_l \log\left(\sum_{k\in\mathcal{L}\setminus\{l\}} \mathbf{x}_k^T\mathbf{W}\mathbf{x}_l g_{kl}p_k + \eta_l\right). \tag{10}$$

Since the first term is independent of the channel assignment vector $\mathbf{x}$, problem (9) reduces to:

$$\text{minimize}_{\mathbf{x}\in\mathcal{X}} \sum_{l\in\mathcal{L}} \lambda_l \log\left(\sum_{k\in\mathcal{L}\setminus\{l\}} \mathbf{x}_k^T\mathbf{W}\mathbf{x}_l g_{kl}p_k + \eta_l\right). \tag{11}$$

3.2. Exclusive scheduling

Most of the existing MAC protocols do not implement the all-at-once scheduling. Instead, they use various carrier sensing mechanisms. The wireless mesh routers compete to access the shared medium. In this case, only one logical link within a neighborhood can be active at a time. Next, we explain how we can take the effect of MAC layer channel access competition into account. In this regard, we first need to clarify the concept of mutual interference.

In an MC–WMN, where only the non-overlapped frequency channels are being used, two links $l, k \in \mathscr{L}$ are defined *mutually interfered* with each other whenever they are assigned to the same channel (i.e., $\mathbf{x}_l^T\,\mathbf{x}_k = 1$) and the sender of one link is within the *interference range* of the receiver of the other link. In this case, links l and k cannot be active simultaneously. The interference range is defined as the region where a given receiver cannot decode the signal correctly if there is another transmission within that range. Given the modulation scheme, the interference range depends on the *minimum required* SINR, which is denoted by $\gamma_{\min}$.

Now consider the case where the frequency channels are partially overlapped. If the interference power of the transmission on link k causes the SINR on link l to be lower than $\gamma_{\min}$, then the transmitter of link k is within the interference range of the receiver node of link l. That is,

$$\frac{g_{ll}p_l}{w_{vu}g_{lk}p_k + \eta_l} < \gamma_{min}. \tag{12}$$

Without loss of generality, we model the path loss g_{kl} using the Friis free space model [33]:

$$g_{kl} = \frac{\alpha}{(e_{kl})^{\kappa}}, \tag{13}$$

where e_{kl} is the Euclidean distance between the transmitter node of link k and the receiver node of link l, κ is the path loss exponent, and α is a constant which depends on the transmitter and the receiver antenna gains and signal wavelength. By substituting (13) into (12) and rearranging the terms, link k interferes with link l if

$$e_{kl} < \sqrt[\kappa]{\left(\frac{\alpha p_k}{g_{ll}p_l/\gamma_{min} - \eta_l}\right) w_{vu}}. \tag{14}$$

The importance of (14) is that we now have different interference ranges depending on the assigned channels to the neighboring links. The smaller the portion of the frequency overlapping, the shorter the interference range will be. Given that the bandwidth and the roll-off factor are the same in all raised cosine channel filters, the interference range only depends on the *frequency channel separation* ($|u - v|$). This fact is illustrated in Fig. 3 where $\delta = 1$. The outermost circle indicates the interference range of the receiver node d when $|u - v| = 0$ (i.e., the same channel is assigned to links k and l). The next circle shows the interference range when $|u - v| = 1$. The innermost circle corresponds to the interference range when $|u - v| = 3$. When $|u - v| > 3$, there is no overlapping between frequency channels m and n for IEEE 802.11b (see Fig. 2a). Thus, the corresponding interference ranges are equal to zero. Note that in this example, transmissions on link l interfere with the transmissions on link k only when either $|u - v| = 0$ or $|u - v| = 1$.

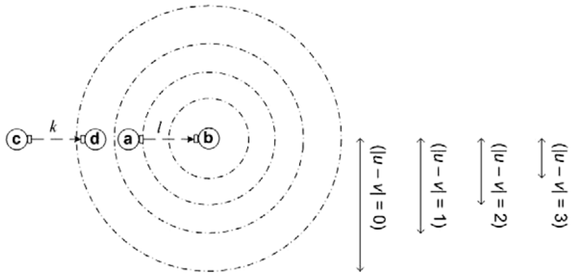


Fig. 3. Different interference ranges depending on the frequency channel separation $|u - v|$. Logical links l and k use channels u and v in 802.11b frequency band, respectively.

For any links $l, k \in \mathscr{L}$, we define a symmetric $C \times C$ *mutual interference matrix* $\mathbf{M}_{lk}$. If either

$$e_{lk} < \sqrt[\kappa]{\left(\frac{\alpha p_l}{g_{kk}p_k/\gamma_{min} - \eta_k}\right) w_{uv}}, \tag{15}$$

or

$$e_{kl} < \sqrt[\kappa]{\left(\frac{\alpha p_k}{g_{ll}p_l/\gamma_{min} - \eta_l}\right) w_{vu}}, \tag{16}$$

then the entry in the uth row and the vth column of $\mathbf{M}_{lk}$ is equal to 1; otherwise, it is equal to 0. If the transmission powers are fixed, the mutual interference matrices are constant for stationary MC–WMNs. For the scenario in Fig. 3, the corresponding mutual interference matrices are *tridiagonal* with all diagonal, subdiagonal, and superdiagonal entries equal to one:

$$\mathbf{M}_{lk} = \mathbf{M}_{kl} = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 & 0 \\ 0 & \cdots & 0 & 1 & 1 & 1 \\ 0 & \cdots & 0 & 0 & 1 & 1 \end{bmatrix}_{11\times 11}. \tag{17}$$

Note that if the logical links l and k are far enough from each other, then all entries of $\mathbf{M}_{lk}$ become zero. According to the definitions of the channel assignment vector and the mutual interference matrix, links l and k cannot be active simultaneously if $\mathbf{x}_l^T\,\mathbf{M}_{lk}\,\mathbf{x}_k = 1$. From this, we define the *opponent set* of link l as follows:

$$\mathscr{O}_l(\mathbf{x}) = \left\{k : k \in \mathscr{L} \setminus \{l\},\ \mathbf{x}_l^T\,\mathbf{M}_{lk}\,\mathbf{x}_k = 1\right\}, \quad \forall\, l \in \mathscr{L}. \tag{18}$$

The cardinality of the set $\mathscr{O}_l$ is denoted by O_l. Since the mutual interference matrix $\mathbf{M}_{lk}$ is symmetric, link $k \in \mathscr{O}_l(\mathbf{x})$, if and only if link $l \in \mathscr{O}_k(\mathbf{x})$. Let q_l denote the *persistent probability* of logical link $l \in \mathscr{L}$. That is, at each time slot $t \in \mathscr{T}$, link l is active with probability q_l. The persistent probabilities can be obtained directly from the MAC protocols which are being used (cf. [38,39]). Considering the IEEE 802.11 DCF, the persistent probabilities can be calculated according to the size of the contention windows [40,41]. In general, regardless of the employed MAC protocol, each node $n \in \mathcal{N}$ can *measure* the persistent probability q_l for all of its outgoing links $l \in \mathcal{L}_n^{\text{out}}$. For notation simplicity, we stack up the persistent probabilities of all logical links and denote the obtained $L \times 1$ vector by $\mathbf{q}$. According to the *exclusive scheduling* model, the transmissions on any logical link $l \in \mathcal{L}$ is successful if no other link in the opponent set $\mathcal{O}_l$ is active at the same time. This happens with probability [42]:

$$q_l \prod_{k \in \mathcal{O}_l(\mathbf{x})} (1 - q_k) = q_l \prod_{k \in \mathcal{L} \setminus \{l\}} (1 - \mathbf{x}_l^T \mathbf{M}_{lk} \mathbf{x}_k q_k), \tag{19}$$

where

$$1 - \mathbf{x}_l^T \mathbf{M}_{lk} \mathbf{x}_k q_k = \begin{cases} 1 - q_k, & \text{if } k \in \mathcal{O}_l(\mathbf{x}), \\ 1, & \text{otherwise.} \end{cases} \tag{20}$$

Logical link $l \in \mathcal{L}$ and any link $k \in \mathcal{L} \setminus (\mathcal{O}_l(\mathbf{x}) \cup \{l\})$ do not mutually interfere with each other. They can be active simultaneously. However, we should still consider the effect of their interference power on each other. In fact, assuming that link l is active and no other link $k \in \mathcal{O}_l(\mathbf{x})$ is active, the average interference power on link l can be obtained as:

$$\begin{aligned} \sum_{k \in \mathcal{L} \setminus (\mathcal{O}_l(\mathbf{x}) \cup \{l\})} q_k \mathbf{x}_k^T \mathbf{W} \mathbf{x}_l g_{kl} p_k &= \sum_{k \in \mathcal{L} \setminus \{l\}} q_k (1 - \mathbf{x}_k^T \mathbf{M}_{kl} \mathbf{x}_l)(\mathbf{x}_k^T \mathbf{W} \mathbf{x}_l) g_{kl} p_k \\ &= \sum_{k \in \mathcal{L} \setminus \{l\}} q_k \mathbf{x}_k^T ((\mathbf{1} - \mathbf{M}_{kl}) \circ \mathbf{W}) \mathbf{x}_l g_{kl} p_k, \end{aligned} \tag{21}$$

where $\circ$ denotes the Hadamard product[2] and $\mathbf{1}$ is a $C \times C$ matrix with all entries equal to 1. The entry in the uth row and vth column of matrix $(\mathbf{1} - \mathbf{M}_{kl}) \circ \mathbf{W}$ is equal to w_{uv} if logical links l and k are *not* mutually interfered over channels u and v; otherwise, it is equal to zero. From (4), (19), and (21), we can obtain the average link capacities to be as follows:

$$\begin{aligned} c_l(\mathbf{x}) = \frac{q_l}{T_S} &\left(\prod_{k \in \mathcal{L} \setminus \{l\}} (1 - \mathbf{x}_l^T \mathbf{M}_{lk} \mathbf{x}_k q_k) \right) \\ &\times \log \left(1 + (K g_{ll} p_l) \Big/ \left(\sum_{k \in \mathcal{L} \setminus \{l\}} q_k \mathbf{x}_k^T ((\mathbf{1} - \mathbf{M}_{kl}) \circ \mathbf{W}) \mathbf{x}_l g_{kl} p_l + \eta_l \right) \right), \\ &\forall l \in \mathcal{L}. \end{aligned} \tag{22}$$

As the minimum required SINR tends to zero (i.e., $\gamma_{\min} \to 0$), for all links $l \in \mathcal{L}$, the opponent set $\mathcal{O}_l$ becomes an empty set and $q_l \to 1$. That is, all links can be always active simultaneously. We would also have $\mathbf{M}_{lk} = \mathbf{0}$ for any $l \in \mathcal{L}$ and each $k \in \mathcal{L} \setminus \{l\}$. In fact, the capacity model in (8) is a special case of the capacity model in (22). Similar to (9), we can replace $c_l(\mathbf{x})$ in problem (CACA) with (22) and obtain the complete formulation of our congestion-aware channel assignment model. We also notice that for all links $l, k \in \mathcal{L}$, the matrices $\mathbf{W}, \mathbf{M}_{lk}$, and $(\mathbf{1} - \mathbf{M}_{kl}) \circ \mathbf{W}$ are constant and independent of $\mathbf{x}$. Thus, they can be obtained off-line and later be used in the corresponding algorithm implementations. Next, we propose a distributed algorithm to solve the congestion-aware channel assignment problem (CACA).

4. Distributed congestion-aware channel assignment (DCACA) algorithm

Since the optimization variables are binary, problem (CACA) is a *combinatorial* problem and is *NP*-hard [44]. It can be solved in a centralized manner. In this regard, the congestion measures for all links (i.e., $\boldsymbol{\lambda} = (\lambda_l, \forall l \in \mathcal{L})$) need to be gathered every T_G time slots (see Section 2) in a pre-authorized node (e.g., one of the gateways). The pre-authorized node then solves problem (CACA) and announces the selected optimal channels to all other nodes. In that case, the pre-authorized node should solve a combinatorial problem with C^L combinations. This may not be tractable when the network grows in size and the number of logical links increases.

In this section, we propose a distributed algorithm to obtain a near optimal solution of problem (CACA) with low complexity. In this regard, each node $n \in \mathcal{N}$ is responsible for assigning the optimal channels only to a subset of links. Recall from Section 2 that we assume the logical topology to be *ripple-effect free* [16,15,17]. Two sample ripple-effect free MC–WMN logical topologies are *Hyacinth* [15] and *TiMesh* [17], which are shown in Fig. 1a and b, respectively. In a ripple-effect free logical topology, there exists an *exclusive* (i.e., not shared) NIC in *at least* one end of *each* logical link. For each logical link $l \in \mathcal{L}$, if it shares an NIC on node n (i.e., if there exists $i \in \mathcal{I}_n$ such that $l \in \mathcal{L}_{n,i}$ and $|\mathcal{L}_{n,i}| > 1$), then we define $s_l = n$. If link l is between nodes n and m and it does not share an NIC on nodes n and m (i.e., if there exist $i \in \mathcal{I}_n$ and $j \in \mathcal{I}_m$ such that $l \in \mathcal{L}_{n,i} \cap \mathcal{L}_{m,j}, |\mathcal{L}_{n,i}| = 1$, and $|\mathcal{L}_{m,j}| = 1$), then we arbitrarily choose either $s_l = n$ or $s_l = m$. In our model, for each link $l \in \mathcal{L}$, wireless node s_l is in charge of the channel assignment. Recall from Section 2 that each logical link uses an exclusive NIC in at least one end. Whenever node s_l assigns a new channel to link l, no further channel assignment is required in any other node. Thus, the wireless nodes can independently assign the frequency channels of their corresponding logical links. For the sample MC–WMN topology in Fig. 1a, we have: $s_{l_1} = s_{l_2} = s_{l_3} = s_{l_4} = a$ and $s_{l_5} = s_{l_6} = s_{l_7} = s_{l_8} = c$. On the other hand, in Fig. 1b, $s_{l_3} = s_{l_4} = s_{l_9} = s_{l_{10}} = s_{l_{11}} = s_{l_{12}} = s_{l_{13}} = s_{l_{14}} = d$.

For each node $n \in \mathcal{N}$, we define:

$$\mathbf{x}_{-n} = (\mathbf{x}_l, \ \forall l \in \mathcal{L}, \ s_l \neq n). \tag{23}$$

That is, $\mathbf{x}_{-n}$ denotes the vector of channel assignment variables corresponding to all logical links *other than* those links that wireless node n is responsible for their channel assignment. Given an arbitrary channel assignment vector $\hat{\mathbf{x}}_{-n}$, we also define:

$$\mathcal{X}_n(\hat{\mathbf{x}}_{-n}) = \{\mathbf{x} : \ \mathbf{x} \in \mathcal{X}, \ \mathbf{x}_{-n} = \hat{\mathbf{x}}_{-n}\}. \tag{24}$$

[2] The Hadamard product of two $C \times C$ matrices $\mathbf{A}$ and $\mathbf{B}$ is a $C \times C$ matrix whose entry in the uth row and vth column is equal to the product of the entry in the uth row and vth column of $\mathbf{A}$ and the entry in the uth row and vth column of $\mathbf{B}$ [43].

That is, $\mathscr{X}_n(\hat{\mathbf{x}}_{-n})$ denotes the set of feasible channel assignment vectors for all logical links $l \in \mathscr{L}$ such that $s_l = n$, assuming that fixed channels are assigned to the rest of the logical links according to $\hat{\mathbf{x}}_{-n}$. In our distributed algorithm, each wireless node n, which is responsible to assign the frequency channels to at least one logical link (i.e., $\exists\, l \in \mathscr{L}$ such that $s_l = n$), solves the following *local* congestion-aware channel assignment problem:

$$\text{maximize}_{\mathbf{x} \in \mathscr{X}_n(\hat{\mathbf{x}}_{-n})} \sum_{l \in \mathscr{L}} \lambda_l c_l(\mathbf{x}), \qquad \text{(LOCAL-CACA)}$$

where the entries of $\hat{\mathbf{x}}_{-n}$ are informed to node n by other nodes $m \in \mathscr{N} \setminus \{n\}$. Let $L^{\max} = \max_{n \in \mathscr{N}} |\mathscr{L}_n|$. It is clear that for all $n \in \mathscr{N}$, problem (LOCAL-CACA) is a combinatorial problem with *at most* $C^{L^{\max}}$ combinations. As the network grows in size, L increases monotonically while $L^{\max}$ is almost fixed. In practice, $L^{\max} \ll L$. Thus, solving the local problem in (LOCAL-CACA) is significantly less complicated compared to solving the global problem in (CACA).

Our proposed distributed congestion-aware channel assignment (DCACA) algorithm is shown in Algorithm 1. For each node $n \in \mathscr{N}$, we define $\mathscr{T}_{L,n} \subset \mathscr{T}$ to denote the set of time slots at which problem (LOCAL-CACA) is solved in node n. Any two consecutive members of set $\mathscr{T}_{L,n}$ are assumed to be T_L time units different. The wireless nodes solve problem (LOCAL-CACA) *asynchronously*. That is, for all $n, m \in \mathscr{N}$, we have $\mathscr{T}_{L,n} \cap \mathscr{T}_{L,m} = \{\}$. In lines 2–5 of Algorithm 1, we initialize the algorithm parameters. In particular, we initially set the network to operate on the first channel in a single-channel scenario. This is required to make sure that all nodes can primarily communicate with each other to form the logical topology. The logical topology formation in line 5 can be performed similar to the steps explained in either [15] or [17]. Every T_G time slots (see Section 2) and in lines 7 and 8, each node measures the congestion level at its outgoing links and broadcasts the results to the rest of the network. By running lines 12 to 18, each node independently solves problem (LOCAL-CACA) using exhaustive search. After assigning the optimal channels in line 20, node n informs the new channels to other nodes.

We are now ready to show the following result:

Theorem 1. *If $T_L \ll T_G$, Algorithm 1 converges to a local optimum of problem (CACA).*

Proof. At any time slot $t \in \mathscr{T}$, we define:

$$\psi(t) = \sum_{l \in \mathscr{L}} \lambda_l(t) c_l(\mathbf{x}(t)). \qquad (25)$$

That is, $\psi(t)$ denotes the objective function of problem (CACA) at time t. We first notice that $\psi(t)$ is always bounded. In particular, we have:

$$\psi(t) \geqslant 0, \quad \forall t \in \mathscr{T} \qquad (26)$$

and

$$\psi(t) \leqslant \frac{1}{T_S} \log\left(1 + \frac{K g^{\max} p^{\max}}{\eta^{\min}}\right)(L\lambda^{\max}), \quad \forall t \in \mathscr{T}, \qquad (27)$$

where $g^{\max} = \max_{l \in \mathscr{L}} g_{ll}, p_l = \max_{l \in \mathscr{L}} p_l, \eta^{\min} = \min_{l \in \mathscr{L}} \eta_l$, and $\lambda^{\max} = \max_{l \in \mathscr{L},\, t \in \mathscr{T}} \lambda_l(t)$. Note that at each time slot $t \in \mathscr{T}, \sum_{l \in \mathscr{L}} \lambda_l(t) \leqslant (L\lambda^{\max})$ and for all logical links $l \in \mathscr{L}$, we have: $c_l(\mathbf{x}(t)) \leqslant \frac{1}{T_S} \log(1 + (K g^{\max} p^{\max})/\eta^{\min})$. We also notice that for any $t_G \in \mathscr{T}_G$, the scalar function $\psi(t)$ is *non-decreasing* during time slots $[t_G + 1, t_G + T_G]$. To show this, consider an arbitrary $t \in [t_G + 1, t_G + T_G]$. If $t \notin \cup_{n \in \mathscr{N}} \mathscr{T}_{L,n}$, then $\psi(t+1) = \psi(t)$. On the other hand, if there exists a wireless node $n \in \mathscr{N}$ such that $t \in \mathscr{T}_{L,n}$, then $\mathbf{x}(t+1)$ is assigned as the optimal solution of problem (LOCAL-CACA). Thus, $\mathbf{x}(t+1)$ is different from $\mathbf{x}(t)$ only if $\psi(t+1)$ is greater than $\psi(t)$. Otherwise, $\mathbf{x}(t+1) = \mathbf{x}(t)$ and $\psi(t+1) = \psi(t)$. Knowing that $\psi(t)$ is bounded and non-decreasing, the convergence of Algorithm 1 is guaranteed as long as T_G is large enough. Let $\mathbf{x}^{\star} \in \mathscr{X}$ denote a *stationary point* of Algorithm 1. Also let $\psi^{\star} \in [0, 1/T_S(L\lambda^{\max}) \log(1 + K g^{\max} p^{\max}/\eta^{\min})]$ denote the value of the objective function of problem (CACA) at stationary point $\mathbf{x}^{\star}$. We first assume that $\mathbf{x}^{\star}$ is *not* a local optimal solution of problem (CACA). Since the objective functions in problems (CACA) and (LOCAL-CACA) are the same, there should exist at least one wireless node $n \in \mathscr{N}$ such that it can partially deviate$\mathbf{x}^{\star}$ to increase $\psi(t)$; however, this contradicts the fact that $\mathbf{x}^{\star}$ is a stationary point. Thus, $\mathbf{x}^{\star}$ and $\psi^{\star}$ represent a local optimal solution and a local optimum of problem (CACA), respectively. □

Algorithm 1 – Distributed congestion-aware channel assignment (DCACA): To be executed by each wireless mesh router $n \in \mathscr{N}$.

1: Allocate memory for $\mathbf{x}^{\star}, \psi^{\star}, \boldsymbol{\lambda}, \mathbf{q}$, and $\hat{\mathbf{x}}_{-n}$.
2: Set $\lambda = \mathbf{1}$.
3: Set $\mathbf{q} = \mathbf{1}$.
4: Set $\psi^{\star} = 0$.
5: Set $\mathbf{x}^{\star} = [[1 \quad 0 \quad \cdots \quad 0] \cdots [1 \quad 0 \quad \cdots \quad 0]]^T$.
6: Form the logical topology using the topology formation algorithms proposed in [15] or [17].
7: **for** all $t \in \mathscr{T}$ **do**
8:   **if** $t \in \mathscr{T}_G$ **then**
9:     Set $\lambda_n = (\lambda_l,\ \forall\, l \in \mathscr{L}_n^{\text{out}})$ according to the congestion measurements.
10:     Set $\mathbf{q}_n = (q_l,\ \forall\, l \in \mathscr{L}_n{}^{\text{out}})$ according to the persistent probability measurements.
11:     Inform λ_n and $\mathbf{q}_n$ to all nodes $m \in \mathscr{N} \setminus \{n\}$.
12:     Set $\psi^{\star} = 0$.
13:   **end if**
14:   **if** $t \in \mathscr{T}_{L,n}$ **then**
15:     **for** all $\mathbf{x} \in \mathscr{X}_n(\hat{\mathbf{x}}_{-n})$ **do**
16:       Set $c_l(\mathbf{x})$ for all $l \in \mathscr{L}$ according to (22) given $\mathbf{q}$.
17:       Set $\psi = \sum_{l \in \mathscr{L}} \lambda_l c_l(\mathbf{x})$.
18:       **if** $\psi > \psi^{\star}$ **then**
19:         Set $\mathbf{x}^{\star} = \mathbf{x}$.
20:         Set $\psi^{\star} = \psi$.
21:       **end if**
22:     **end for**
23:     Assign the frequency channels according to $\mathbf{x}^{\star}$.
24:     Inform $\hat{\mathbf{x}}_n = (\mathbf{x}_l^{\star},\ \forall\, l \in \mathscr{L}_n,\ s_l = n)$ to all nodes $m \in \mathscr{N} \setminus \{n\}$.
25:   **end if**
26: **end for**

From Theorem 1, if problem (CACA) only has a unique local optimal solution, then Algorithm 1 will indeed converge to the best possible frequency channel assignment according to problem (CACA). In many other cases, although there are more than one local optimal solutions, they are equally good. For example, assume that $C = 2$ and $L = 3$. In this case, $\mathbf{x} = [[1 \quad 0][1 \quad 0][0 \quad 1]]^T$ and $\mathbf{x} = [[0 \quad 1][0 \quad 1][1 \quad 0]]^T$ result in the same performance. In fact, in both cases, the first and the second logical links operate over the same frequency channel while the third link operates on a different channel. We will further investigate the optimality of Algorithm 1 and its due effect on network performance in Section 5.

We also notice that based on Algorithm 1, the wireless nodes are *not* selfish. In fact, they *cooperate* with each other. This is indeed necessary for achieving the optimal network performance in a distributed fashion. Assuming the case where each node $n \in \mathcal{N}$ acts selfishly, it would solve the following *selfish* local problem:

$$\text{maximize}_{\mathbf{x} \in \mathcal{X}_n(\hat{\mathbf{x}}_{-n})} \sum_{l \in \mathcal{L}_n} \lambda_l c_l(\mathbf{x}), \qquad \text{(SELFISH-CACA)}$$

where only the links of node n (i.e., the links in set $\mathcal{L}_n$) are taken into account. By solving problem (SELFISH-CACA), node n would not take into account the interference that its transmissions cause on the transmissions from other nodes. This selfish behavior has been seen in various proposed channel assignment strategies in the literature. For example, according to the *Load-Aware* channel assignment strategy in [15], each node assigns its links with the frequency channels which are *less used* by its neighboring transmissions, i.e., the *best* available channels. However, in our proposed strategy, some nodes may reserve a channel for a highly congested logical link to help it to resolve its congestion problem. As we will show in Section 5, the cooperation in Algorithm 1 makes it noticeably superior compared to the *Load-Aware* algorithm.

5. Performance evaluation

In this section, we assess the performance of our proposed DCACA algorithm based on *ns-2* simulations [45]. To support multiple NICs on each wireless node, the *ns*-2 patch from [46] is being used. We modified the patch so that it can also support the partially-overlapped frequency channels. In the simulation model, the size of the network field is 1000 m × 1000 m. Unless we specify otherwise, the MC–WMN consists of 60 wireless mesh routers (i.e., $N = 60$). Four of them serve as the gateways, and they are located at the four corners in the field.

Ten different topologies are generated. Topology numbers $1, 2, \ldots, 5$ correspond to five different *grid* topologies, whereas topology numbers $6, 7, \ldots, 10$ correspond to five different *random* topologies. For the grid topologies, the size of each grid is 8×8. The distance between two adjacent grid points is 140 m. Nodes are placed in 60 (out of 64) grid points randomly. For the random topologies, nodes are randomly placed in the network field such that each node has at least one neighbor within its communication range. Once the physical topology has been created, the logical topology is formed based on the *Hyacinth* ripple-effect free WMN architecture [15]. We also set $T_G = 60$ s and $T_{L,n} = 5$ s for all $n \in \mathcal{N}$.

In our simulation model, the traffic sources are assumed to be TCP Vegas and TCP Reno. For the case when TCP Vegas sources are used, for each logical link $l \in \mathcal{L}$, the congestion measure λ_l is its queuing delay. On the other hand, for the case when TCP Reno sources are used, for each logical link $l \in \mathcal{L}$, the congestion measure λ_l is the link's packet loss rate (i.e., the probability of dropping a packet). We modified the IEEE 802.11 module in *ns*-2 so that the higher layer applications can access the vector of the queueing delays and packet loss rate $\boldsymbol{\lambda}$. For a given topology, in each simulation run, 30 wireless nodes are randomly selected as either the source (or destination) for TCP flows to (or from) the Internet (i.e., the corresponding gateway). The simulation time is 300 s. For each wireless node, the chosen gateway is the one which corresponds to the minimum hop path.

In our performance evaluation, we consider the following performance metrics: (1) *Aggregate throughput*: total number of correctly received TCP segments (in bits) at the destinations divided by the total simulation time. (2) *Average round-trip time*: average time delay between sending a TCP segment and receiving its acknowledgement. For TCP Vegas sources, we also consider the aggregate network utility as the criterion to evaluate the optimality of our proposed algorithm in terms of solving problem (CACA), where λ_l denotes the queueing delay for each link $l \in \mathcal{L}$. Notice that TCP Vegas sources have *logarithmic* utility functions as shown in [47]. For the case of TCP Reno, although this protocol has been reverse engineered (cf. [30]), finding an accurate utility function is a difficult task. Therefore, we evaluate the optimality of DCACA algorithm only for the case when TCP Vegas sources are being simulated.

The parameters that we used in the simulations are shown in Table 2. Note that most of them are the default *ns-2* parameters. For the CSMA/CA (Carrier Sense Multiple Access) MAC protocol, the RTS/CTS (Request-To-Send/Clear-To-Send) mechanism is enabled. We also considered the case where all the competing links in each neighborhood are assumed to have an equal chance to access the shared medium. That is, $q_l = 1/(1 + O_l)$ for all links $l \in \mathcal{L}$.

Table 2
List of *ns-2* simulation parameters

Transmission power (p_l)	0.2818 W
Communication range	250 m
Carrier sensing range	450 m
Receive threshold	3.652×10^{-10} W
Carrier sensing threshold is	1.559×10^{-11}W
Capture threshold $SINR_{min}$	10.0
Path loss parameter (κ)	9.35×10^{-5}
Thermal noise power (n_l)	1.0×10^{-11} W
IEEE 802.11a data rate	54 Mbps
IEEE 802.11b data rate	11 Mbps
Queue type	Drop-Tail
Queue size	50 Pkts
TCP packet size	1000 Bytes
TCP Vegas alpha parameter	1
TCP Vegas beta parameter	3
TCP Reno slow-start threshold	20

5.1. Comparison with Load-Aware algorithm

We first compare the performance between *DCACA* and *Load-Aware* [15] algorithms. Both algorithms are distributed and run on top of the *Hyacinth* ripple-effect free MC–WMN logical topologies. This leads to an accurate and fair comparison. We assume that there are two IEEE 802.11a NICs in each wireless mesh router (i.e., $I = I_n = 2$ for all $n \in \mathcal{N}$). We first limit our study to the case where all available frequency channels are non-overlapping and only six channels are available (i.e., $C = 6$). We will later consider the case where more channels and NICs are available and also the case where the frequency channels are not only orthogonal, but also partially-overlapped in Section 5.2 and 5.3, respectively.

Fig. 4 shows the aggregate throughput and the average round-trip time for all ten different topologies when TCP Vegas sources are being used. Results from Fig. 4a show that on average, *DCACA* can increase the aggregate throughput by 191.8% compared to the single-channel case, and by 11.5% compared to the multi-channel case where the *Load-Aware* algorithm is used. Results from Fig. 4b also show that *DCACA* can reduce the average round-trip time by 234.1% compared to the single-channel case, and by 35.3% compared to the multi-channel case where the *Load-Aware* algorithm is used. The superiority of *DCACA*, especially on reducing the round-trip time, is evident. The better performance of *DCACA* Algorithm can be explained based on its features. Unlike the *Load-Aware* algorithm, where each node selfishly tries to only improve its own performance, *DCACA* algorithm leads to global cooperation among the nodes (see Section 4). On the other hand, our proposed algorithm uses an accurate link capacity model which takes into account various network parameters such as transmission power, wireless path loss, medium access control, and the frequency response of the channel band-pass filters.

Next, we simulate the case with presence of TCP Reno traffic. Results on the aggregate network throughput and the average round-trip time when TCP Reno sources are being used are shown in Fig. 5. We can see that regardless of the choice of the TCP protocol, our proposed DCACA algorithm can manage to control congestion. Notice that by changing the TCP protocol, DCACA should change the choice of congestion price. In particular, for the case of TCP Reno traffic, DCACA should consider the packet loss rate as the congestion price on each link. Results from Fig. 5a show that on average, *DCACA* can increase the aggregate throughput by 177.2% compared to the single channel case, and by 9.8% compared to the multi-channel case where the *Load-Aware* algorithm is used. Results from Fig. 5b also show that *DCACA* can reduce the average round-trip time by 181.3% compared to the single-channel case, and by 28.7% compared to the multi-channel case where the *Load-Aware* algorithm is used.

From [29], TCP Vegas aims to control the queueing delay along the routing path of each TCP session. In the next experiment, we vary the number of TCP flows and investi-

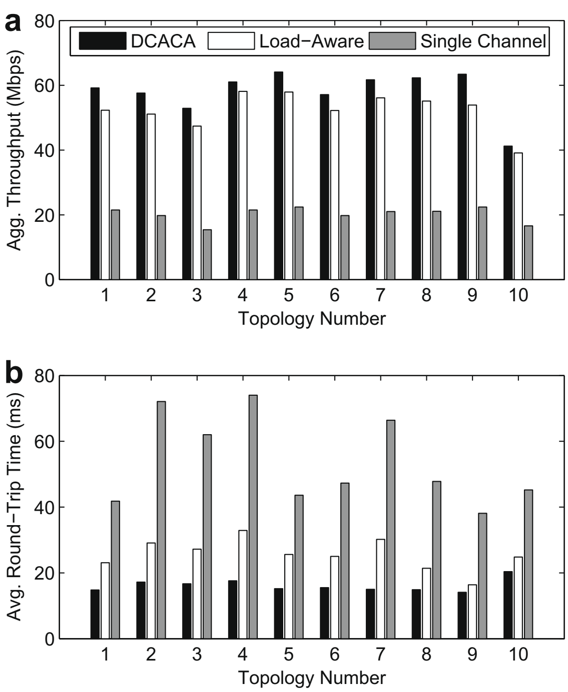


Fig. 4. Performance comparison between *DCACA* and *Load-Aware* distributed channel assignment algorithms in presence of TCP Vegas traffic. (a) Aggregate throughput, (b) Average round-trip time.

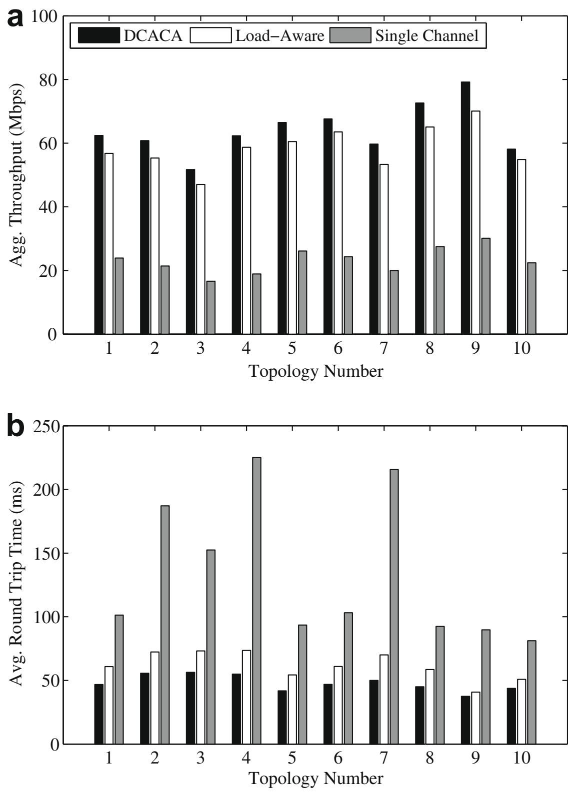


Fig. 5. Performance comparison between *DCACA* and *Load-Aware* distributed channel assignment algorithms in presence of TCP Reno traffic. (a) Aggregate throughput, (b) Average round-trip time.

gate the impact of number of flows on the round-trip time. Fig. 6a and b show the average round-trip time when the grid topology (with topology number 1) and the random topology (with topology number 6) are being simulated, respectively. Simulation results show that by using a single frequency channel, the network becomes highly congested when the number of TCP flows is greater than ten. On the other hand, by using two NICs and assigning six orthogonal frequency channels, the congestion is effectively avoided in all cases as long as our proposed *DCACA* is being used. Note that the *DCACA* algorithm assigns the appropriate channels in the neighborhood to increase the effective capacities on the congested links. It thus prevents any of the logical links to become severely congested avoiding large queuing delays.

5.2. Impact on available resources

There are two important resources in an MC–WMN: the NICs at each wireless mesh router, and the available frequency channels. To evaluate the impact of the network resources, we vary the number of NICs at each router from 2 to 4, and the number of non-overlapped channels from 1 to 12 (IEEE 802.11a frequency band). Fig. 7 shows the aggregate throughput and the average round-trip time for the first random topology when TCP Vegas sources are being used. The results for other random and grid topologies are similar and omitted for brevity. We see that the network performance significantly increases as more resources are being used. The improvements can be interpreted in terms of the capacity models in Section 3. For example, increasing the number of NICs reduces the number of logical links that share a common interface. It removes some of the equality constraints in (2) which consequently expands the feasible set $\mathscr{X}$. On the other hand, increasing the number of frequency channels allows us to assign different channels to near-by links and avoid mutual interference among them. Thus, we can have more links $l, k \in \mathscr{L}$ with $\mathbf{x}_l^T \mathbf{x}_k = 0$.

Comparing the single-channel scenario with a multi-channel case where $C = 12$ and $I = I_n = 4$ for all $n \in \mathscr{N}$, the *DCACA* algorithm can increase the aggregate throughput by a factor of 5.4 and decrease the average round-trip time by a factor of 5.6.

Next, we study the impact of resources when TCP Reno sources are being used. Fig. 8 shows the aggregate throughput and the average round-trip time in this case. Again, we can see that the network performance increases as more resources are being used. We can conclude that regardless of the choice of TCP protocol, our proposed DCACA algorithm can properly utilize the available network resources.

5.3. Performance gain by using partially-overlapped channels

There are 12 non-overlapped channels available in the IEEE 802.11a frequency band which can significantly increase the performance, as discussed in Sections 5.1 and 5.2. However, there are only 3 non-overlapped channels available in the IEEE 802.11b frequency band. A small number of non-overlapped frequency channels can limit the benefits of deploying an MC–WMN. The performance

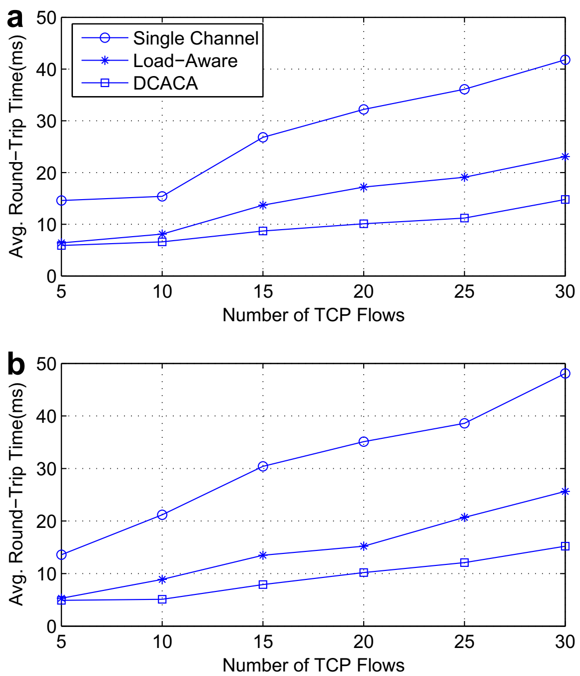


Fig. 6. Average round-trip time versus number of established TCP Vegas flows. (a) Results for topology number 1 (grid topology), (b) Results for topology number 6 (random topology).

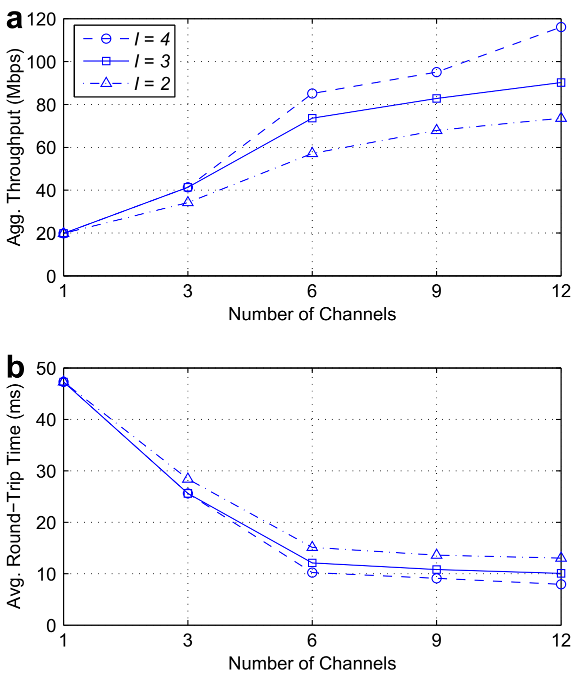


Fig. 7. The performance gain of increasing the number of NICs per node and the number of available orthogonal channels in the first random topology in presence of TCP Vegas traffic. (a) Aggregate network throughput, (b) Average packet round-trip time.

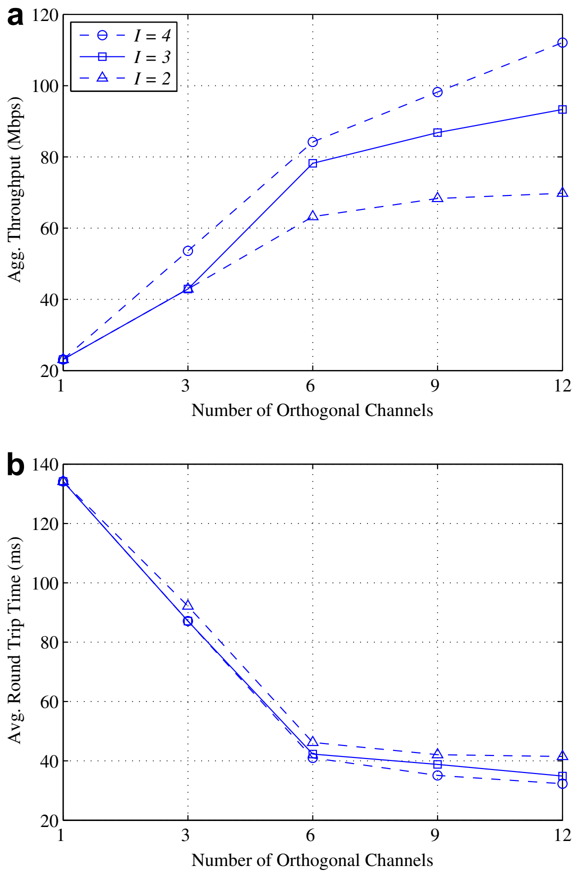


Fig. 8. The performance gain of increasing the number of NICs per node and the number of available orthogonal channels in the first random topology in presence of TCP Reno traffic. (a) Aggregate network throughput, (b) Average packet round-trip time.

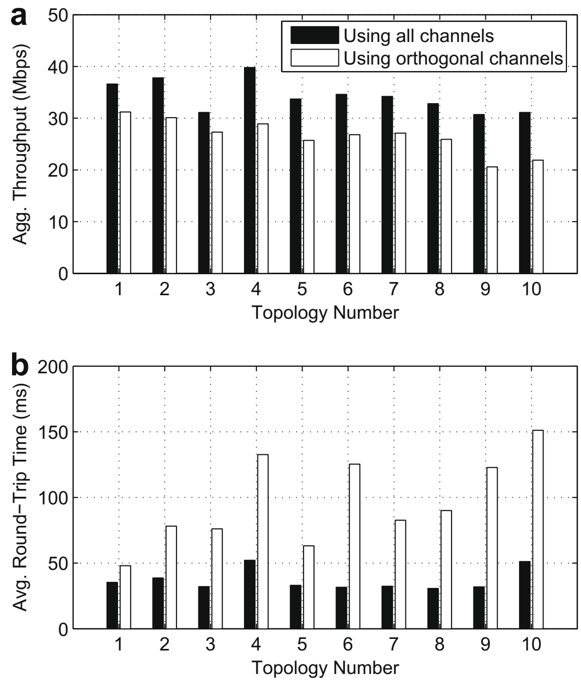


Fig. 9. Performance gain by using all eleven partially-overlapped channels in IEEE 802.11b frequency band instead of only using the three non-overlapped channels 1, 6, and 11 in presence of TCP Reno traffic. There are two interfaces in each wireless router ($I = 2$). (a) Aggregate throughput, (b) Average round-trip time.

can be increased by using all the partially-overlapped channels. In this section, we evaluate the performance gain by using all 11 available channels in comparison with using only three orthogonal channels in the IEEE 802.11b frequency band. We consider the case where there are two NICs in each wireless mesh router. We use the raised cosine filter to model channel band-pass filters and set $\delta = 1$ (see Fig. 2). The aggregate throughput and the average round-trip time when TCP Vegas source are used are shown in Fig. 9a and b, respectively. Note that because of the lower data rate in IEEE 802.11b compared to IEEE 802.11a standard, the round-trip times in Fig. 9b are higher than those in the previous figures. We see that using the partially-overlapped channels is beneficial as it increases the aggregate network throughput by 25% and decreases the average round-trip time by more than one half.

The achieved performance gain is due to an efficient usage of the frequency spectrum. Intuitively, *DCACA* assigns two non-overlapped channels to the near-by congested links, as long as the non-overlapped channels have not been assigned in the neighborhood. Otherwise, it assigns two available partially-overlapped channels that can cause the minimum interference.

Next, we study the performance gain of using partially-overlapped channels in presence of TCP Reno traffic. Results are shown in Fig. 10a. We can see that regardless of the choice of TCP protocol, using all available partially-overlapped channels can improve the performance compared to the case when only the orthogonal channels are being used. Note that because of the lower data rate in IEEE 802.11b compared to IEEE 802.11a standard, the round-trip times in Fig. 9b are higher than those in the previous figures. We see that using the partially-overlapped channels can increase the aggregate network throughput by 27.6% and decreases the average round-trip time by 78.5%.

5.4. Optimality

Recall from Section 2 that solving problem (CACA) helps to solve other resource allocation problems such as network utility maximization and maximum weight matching. The former is an important design objective in the presence of elastic traffic sources. In this section, we evaluate the capability of DCACA in solving the network utility maximization problem across TCP Vegas sources. Unlike TCP Reno sources, the TCP Vegas sources are designed to maximize a specific utility function which is logarithmic [47]. We consider the *aggregate* network utility as the performance metric in this section. Assume that there exists a TCP Vegas source from node $n \in \mathcal{N}$ to node $d \in \mathcal{N} \setminus \{n\}$. Let r_{sd} denote its transmission rate. The utility of this TCP Vegas source is then defined as $D_{sd} \log(r_{sd})$, where D_{sd} de-

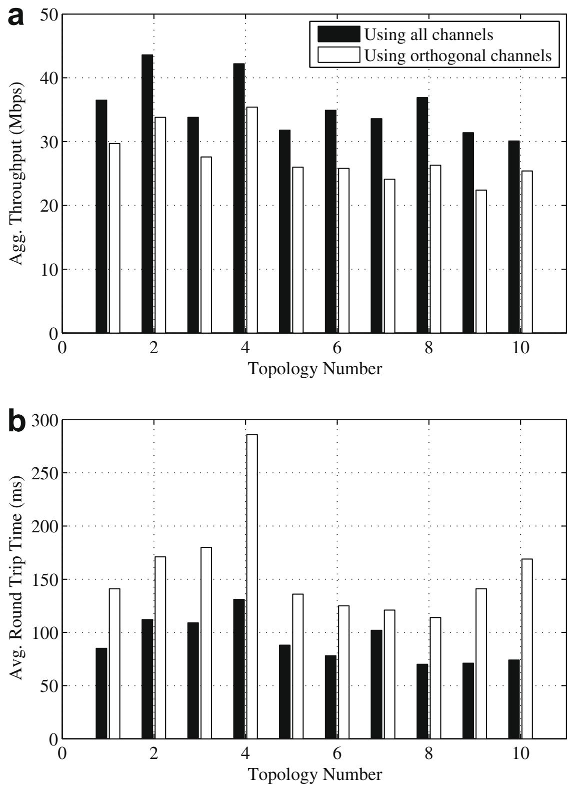


Fig. 10. Performance gain by using all eleven partially-overlapped channels in IEEE 802.11b frequency band instead of only using the three non-overlapped channels 1, 6, and 11 in presence of TCP Reno traffic. There are two interfaces in each wireless router ($I = 2$). (a) Aggregate throughput, (b) Average round-trip time.

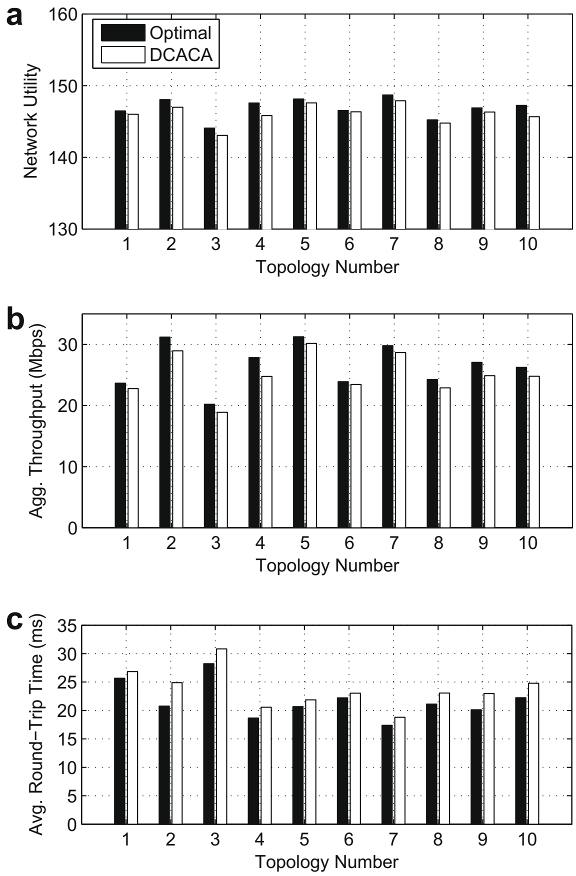


Fig. 11. Optimality of DCACA to solve the network utility maximization problem in presence of TCP Vegas traffic sources where for each link, the corresponding congestion measure is indeed the link's queueing delay. (a) Network Utility, (b) Aggregate throughput, (c) Average round-trip time. We see that DCACA Algorithm results in near optimal network utilities in all cases.

notes the fixed delay at the routing path from node s to node d [47]. The aggregate network utility is then defined as $\sum_{s\in\mathcal{N}}\sum_{d\in\mathcal{N}\setminus\{s\}}D_{sd}\log(r_{sd})$.

To obtain the optimal utility for each topology, we simulate all the feasible channel assignments and select the maximum measured network utility. We consider five random and five grid topologies. Each topology includes 15 nodes (i.e., $N = 15$) and has one gateway. Notice that there is no limitation on running DCACA for large-scale MC–WMNs as we showed in the previous experiments. However, to be able to obtain the exact optimal network utility, we need to examine all the channel assignment possibilities and there are C^L combinations. This required us to limit the network size. Nevertheless, our study here provides a benchmark to evaluate the optimality of DCACA algorithm.

In our simulation model, ten nodes are randomly selected as TCP sources. Each router is equipped with two IEEE 802.11a NICs. We use three orthogonal channels. The rest of the simulation settings are the same as before. Results from Fig. 11a show that our proposed *DCACA* algorithm can lead to 99.4% optimality on average. To have a better understanding on the effect of the optimality gaps on network performance, we have also shown the aggregate network throughput and the average round-trip time in Fig. 11b and c, respectively. On average, using Algorithm 1, the performance degradation on aggregate network throughput and average round-trip time are only 6.3% and 7.9%, respectively. Thus, our proposed distributed congestion-aware channel assignment algorithm can lead to a near optimal solution for the network utility maximization problem. On the other hand, as shown in Section 5.1, our proposed algorithm can significantly improve the network performance compared to the *Load-Aware* distributed channel assignment strategy. In particular, the performance further improves if we use not only the non-overlapped channels, but also the all available partially-overlapped channels.

6. Conclusions and future work

In this paper, we considered the problem of maximizing a weighted summation of all link capacities where for each link, the corresponding weighting parameter is the link's congestion measure. Various congestion measures can be considered such as queueing delay, packet loss rate, or differential backlog. We first obtained a comprehensive closed-form mathematical link capacity model in terms of our defined channel assignment variables. We also introduced the channel overlapping and mutual interference matrices to model the effect of partial frequency overlapping among the channels. Unlike most of the previous channel assignment algorithms, our proposed scheme assigns not only the orthogonal (i.e., non-overlapped) frequency channels, but also the partially-overlapped channels. We then proposed a distributed congestion-aware channel assignment algorithm (DCACA) which works asynchronously among the wireless mesh routers and requires each node only to execute a simple local search procedure. To assess the performance of DCACA algorithm, we performed extensive *ns*-2 network simulations for both grid and random MC–WMN topologies. In the presence of TCP Vegas traffic sources and when the congestion measure of each link is selected to be the corresponding link's queueing delay, our proposed algorithm increases the aggregate throughput by 11.5% and decreases the average packet round-trip time by 35.3% compared to the *Load-Aware* channel assignment algorithm [15]. In the presence of TCP Reno traffic sources and when the congestion measure of each link is selected to be the corresponding link's packet loss rate, our proposed algorithm increases the aggregate throughput by 9.8% and decreases the average packet round-trip time by 28.7% compared to the *Load-Aware* channel assignment algorithm. In a congested IEEE 802.11b network setting, compared with the case where we only used the three non-overlapped channels, the aggregate network throughput can further be increased by 25% and the average round-trip time can be further reduced by more than one half when all the 11 partially-overlapped channels are used.

The current work can be extended in several directions. In particular, we can further consider the impact of using directed antenna to further reduce the interference. In this regard, we can use the interference models in [23] and reformulate the congestion-aware channel assignment problem in (CACA) accordingly. We shall also evaluate our proposed DCACA algorithm through test-bed study. In particular, it is important to assess the algorithm performance in presence of combination of both TCP and UDP traffic. Finally, the proposed joint congestion-aware channel assignment scheme can be further improved by adding power control. Notice that the interference can be reduced not only by assigning distinct channels to neighboring transmissions but also properly adjusting the transmission power of each node.

References

[1] I. Akyildiz, X. Wang, A survey on wireless mesh networks, IEEE Communications Magazine 43 (September) (2005) 23–30.

[2] R. Bruno, M. Conti, E. Gregori, Mesh networks: commodity multihop ad hoc networks, IEEE Communications Magazine 43 (March) (2005) 123–131.

[3] P. Bahl, A. Adya, J. Padhye, A. Wolman, Reconsidering wireless systems with multiple radios, ACM Computer Communications Review 34 (October) (2004) 39–46.

[4] H.S. Chiu, K.L. Yeung, K.S. Lui, On optimization of joint channel assignment and routing in mobile ad hoc networks, in: Proceedings of IEEE Globecom, Washington, DC, 2007.

[5] A.K. Das, H.M.K. Alazemi, R. Vijayakumar, S. Roy, Optimization models for fixed channel assignment in wireless mesh networks with multiple radios, in: Proceedings of IEEE SECON, Santa Clara, CA, 2005.

[6] Y.Y. Chen, S.C. Liu, C. Chen, Channel assignment and routing for multi-channel wireless mesh networks using simulated annealing, in: Proceedings of IEEE Globecom, San Francisco, CA, 2006.

[7] A.H. Mohsenian Rad, V.W.S. Wong, Joint logical topology design, interface assignment, channel allocation, and routing for multi-channel wireless mesh networks, IEEE Transactions on Wireless Communications 6 (December) (2007) 4432–4440.

[8] Y.H. Tam, R. Benkoczi, H.S. Hassanein, S.G. Akl, Optimal channel assignment in multi-hop cellular networks, in: Proceedings of IEEE Globecom, Washington, DC, 2007.

[9] H. Zhou, C. Yeh, H.T. Mouftah, A reliable low-overhead mac protocol for multi-channel wireless mesh networks, in: Proceedings of IEEE Globecom, Washington, DC, 2007.

[10] N.H. Tran, C.S. Hong, Joint scheduling and channel allocation in wireless mesh networks, in: Proceedings of IEEE CCNC, Las Vegas, NV, 2008.

[11] W. Li, P. Fan, K.B. Letaief, A combination scheme of topology control and channel assignment in wireless ad hoc networks, in: Proceedings of IEEE Globecom, Washington, DC, 2007.

[12] M. Alicherry, R. Bhatia, L.E. Li, Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks, IEEE Journal on Selected Areas in Communications 24 (November) (2006) 1960–1971.

[13] M. Kodialam, T. Nandagopal, Characterizing the capacity region in multi-radio multi-channel wireless mesh networks, in: Proceedings of ACM MobiCom, Cologne, Germany, 2005.

[14] Y. Song, C. Zhang, Y. Fang, Throughput maximization in multi-channel wireless mesh access networks, in: Proceedings of IEEE ICNP, Beijing, China, 2007.

[15] A. Raniwala, T. Chiueh, Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network, in: Proceedings of IEEE Infocom, Miami, Florida, 2005.

[16] A. Raniwala, K. Gopalan, T. Chiueh, Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks, ACM Mobile Computing and Communication Review 8 (April) (2004) 50–65.

[17] A.H. Mohsenian Rad, V.W.S. Wong, Logical topology design and interface assignment for multi-channel wireless mesh networks, in: Proceedings of IEEE Globecom, San Francisco, CA, 2006.

[18] G. Zeng, B. Wang, Y. Ding, L. Xiao, M. Mutka, Multicast algorithms for multi-channel wireless mesh networks, in: Proceedings of IEEE ICNP, Beijing, China, 2007.

[19] A. Adya, P. Bahl, J. Padhye, A. Wolman, L. Zhou, A multi-radio unification protocol for IEEE 802.11 wireless networks, in: Proceedings of IEEE Broadnet'04, San Jose, CA, 2004.

[20] S. Sharma, N. Zhu, T. Chiueh, Low-latency mobile IP handoff for infrastructure-mode wireless LANs, IEEE Journal on Selected Areas in Communications 22 (May) (2004) 643–652.

[21] A. Mishra, V. Shrivastava, S. Banerjee, W. Arbaugh, Partially overlapped channels not considered harmful, in: Proceedings of ACM SIGMetric, Saint Malo, France, 2006.

[22] A.H. Mohsenian Rad, V.W.S. Wong, Joint optimal channel assignment and congestion control in multi-radio wireless mesh networks, in: Proceedings of IEEE ICC, Istanbul, Turkey, 2006.

[23] B. Gavish, Y. Ofek, R. Whitaker, Uplink analysis with mobile devices using directional-smart antennas, International Journal of Mobile Network Design and Innovation 1 (2005) 76–83.

[24] M. Felegyhazi, M. Cagalj, S.S. Bidokhti, J.P. Hubaux, Non-cooperative multi-radio channel allocation in wireless networks, in: Proceedings of IEEE INFOCOM, Anchorage, Alaska, 2007.

[25] F. Kelly, Charging and rate control for elastic traffic, European Transactions on Telecommunication 8 (1997) 33–37.

[26] M. Chiang, S.H. Low, A.R. Calderbank, J.C. Doyle, Layering as optimization decomposition: A mathematical theory of network architectures, in: Proceedings of IEEE, vol. 95, 2007, pp. 255–312.

[27] L. Chen, S.H. Low, J. Doyle, Joint congestion control and media access control design for ad hoc wireless networks, in: Proceedings of IEEE Infocom, Miami, Florida, 2005.

[28] J. Wang, L. Li, S.H. Low, J. Doyle, Cross-layer optimization in TCP/IP networks, IEEE/ACM Transactions on Networking 13 (June) (2005) 582–595.

[29] L.S. Brakmo, L.L. Peterson, TCP Vegas: end to end congestion avoidance on a global internet, IEEE Journal on Selected Areas in Communications 13 (October) (1995) 1465–1480.

[30] S. Low, F. Paganini, J. Doyle, Internet congestion control, IEEE Control Systems Magazine (February) (2002) 28–43.
[31] L. Tassiulas, A. Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, IEEE Transactions on Automatic Control 13 (June) (1992) 582–595.
[32] M.J. Neely, E. Modiano, C.E. Rohrs, Dynamic power allocation and routing for time-varying wireless networks, IEEE Journal on Selected Areas in Communications 23 (January) (2005) 89–103.
[33] J. Proakis, Data Communications, fourth ed., McGraw-Hill, 2000.
[34] D. Tse, P. Viswanath, Fundamentals of Wireless Communication, Cambridge University Press, 2004.
[35] M. Chiang, To layer or not to layer: balancing transport and physical layers in wireless multihop networks, in: Proceedings of IEEE Infocom, Hong Kong, China, 2004.
[36] Y. Xi, E.M. Yeh, Throughput optimal distributed control of stochastic wireless networks, in: Proceedings of the Fourth International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'06), Boston, MA, 2006.
[37] K. Karakayali, J. Kang, M. Kodialam, K. Balachandran, Joint resource allocation and routing for ofdma-based broadband wireless mesh networks, in: Proceedings of IEEE ICC, Glasgow, Scotland, 2007.
[38] J. Lee, M. Chiang, R. Calderbank, Utility-optimal random-access control, IEEE Transactions on Wireless Communications 25 (August) (2007) 1135–1147.
[39] A.H. Mohsenian Rad, J. Huang, M. Chiang, V.W.S. Wong, Utility-optimal random access: Reduced complexity, fast convergence, and robust performance, IEEE Transactions on Wireless Communications 8 (February) (2009) 898–911.
[40] Y.C. Tay, K.C. Chua, A capacity analysis for the IEEE 802.11 MAC protocol, Wireless Networks 7 (March) (2001) 159–171.
[41] J. Lee, A. Tang, J. Huang, M. Chiang, A. Calderbank, Reverse engineering MAC: a game-theoretic model, IEEE Journal on Selected Areas in Communications 6 (July) (2007) 2741–2751.
[42] D.P. Bertsekas, R. Gallager, Data Communications, second ed., Prentice Hall, 1992.
[43] R. Horn, C. Johnson, Topics in Matrix Analysis, first ed., Cambridge University Press, 1994.
[44] H. Taha, Operations Research: An introduction, seventh ed., Prentice Hall, 2003.
[45] The ns2 network simulator. <http://www.isi.edu/nsnam/ns/>.
[46] ns2 patch. <http://www.ecsl.cs.sunysb.edu/multichannel/code/ns-2.1b9a-multi-nic-patch>, released 2004.
[47] S.H. Low, L. Peterson, L. Wang, Understanding vegas: a duality model, Journal of the ACM 49 (March) (2002) 207–235.

Amir-Hamed Mohsenian-Rad received his B.Sc. degree from Amir-Kabir University of Technology (Tehran, Iran) in 2002, M.Sc. degree from Sharif University of Technology (Tehran, Iran) in 2004, and Ph.D. degree from the University of British Columbia (Vancouver, Canada) in 2008, all in electrical engineering. From March to July 2007, he was also a visiting scholar at Princeton University (Princeton, NJ). As a graduate student, he received the UBC Graduate Fellowship as well as the Pacific Century Graduate Scholarship from the British Columbia Provincial Government. He currently serves as TPC member for the IEEE International Conference on Communications (ICC09) and the IEEE Consumer Communications and Networking Conference (CCNC09). His research interests are in the area of optimization theory and its applications in computer communications and wireless networking.

Vincent W.S. Wong received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently an Associate Professor. His research interests are in resource and mobility management for wireless mesh networks, wireless sensor networks, and heterogeneous wireless networks. He is an associate editor of the IEEE Transactions on Vehicular Technology and an editor of KICS/IEEE Journal of Communications and Networks. He serves as TPC member in various conferences, including IEEE Globecom, ICC, and Infocom. He is a senior member of the IEEE and a member of the ACM.