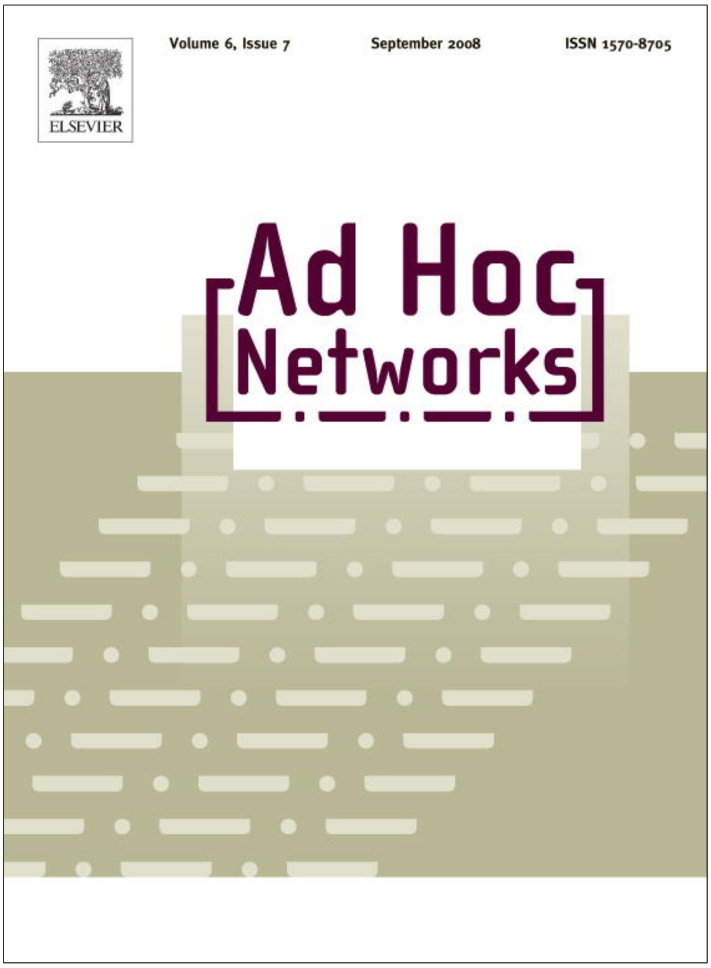
Volume 6, Issue 7
September 2008
ISSN 1570-8705
ELSEVIER
Ad Hoc Networks

Available online at www.sciencedirect.com

ScienceDirect

Ad Hoc Networks 6 (2008) 1051–1077

Ad Hoc Networks

www.elsevier.com/locate/adhoc

# Protocols and architectures for channel assignment in wireless mesh networks

Jorge Crichigno [a,*], Min-You Wu [b,1], Wei Shu [a,2]

[a] *Department of Electrical and Computer Engineering, The University of New Mexico, Albuquerque, NM 87131-0001, United States*
[b] *Department of Computer Science and Engineering, Shanghai JiaoTong University, Shanghai 200240, China*

Received 21 March 2007; received in revised form 9 October 2007; accepted 20 October 2007
Available online 30 October 2007

## Abstract

The use of multiple channels can substantially improve the performance of wireless mesh networks. Considering that the IEEE PHY specification permits the simultaneous operation of three non-overlapping channels in the 2.4 GHz band and 12 non-overlapping channels in the 5 GHz band, a major challenge in wireless mesh networks is how to efficiently assign these available channels in order to optimize the network performance. We survey and classify the current techniques proposed to solve this problem in both single-radio and multi-radio wireless mesh networks. This paper also discusses the issues in the design of multi-channel protocols and architectures.

*Keywords:* Wireless mesh network; Channel assignment; Multi-channel protocol; Medium access control layer

## 1. Introduction

IEEE 802.11-based wireless mesh networks (WMNs) consist of mesh routers and mesh clients equipped with IEEE 802.11 radio interfaces,[3] where mesh routers have minimal mobility and provide wireless access to clients. Each node in a WMN can operate not only as a host but also as a router, forwarding packets on behalf of other nodes that may not be within direct wireless transmission range of their destinations, producing a wireless multi-hop environment [1].

Considering that the IEEE PHY specification [2] permits the simultaneous operation of three non-overlapping channels in the 2.4 gigahertz (GHz) band and 12 non-overlapping channels in the 5 GHz band, the use of multiple channels can improve the capacity of WMNs [3]. Multi-channel protocols and architectures are designed to exploit the available channels to enhance the overall throughput. To achieving this goal, there are two major challenges: (i) the channel assignment

---

[*] Corresponding author. Tel.: +1 505 277 1372.
*E-mail addresses:* jcrichigno@ece.unm.edu (J. Crichigno), mwu@sjtu.edu.cn (M.-Y. Wu), shu@ece.unm.edu (W. Shu).

[1] Tel.: +86 21 34204478.

[2] Tel.: +1 505 277 8471; fax: +1 505 277 1439.

[3] Radio interface, radio and interface will be used as synonymous in this survey.

1570-8705/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2007.10.002

mechanism, which involves assigning channels to interfaces; and (ii) the routing mechanism, which involves routing packets. Although an ideal approach might consider both channel assignment and routing simultaneously, investigating the problem separately could reduce the complexity and prompt the conceptual understanding of issues. This paper focuses on the channel assignment problem and provides a literature survey on architectures and protocols proposed to solve the problem. It also discusses the impact of channel assignment on routing as well as other relevant issues. The survey provides an analysis of a very specific problem – channel assignment – in WMNs that was briefly discussed in the survey by Akyildiz et al. [1].

The remainder of the paper is organized as follows: Section 2 presents an overview of multi-channel wireless mesh networks. Section 3 discusses the issues found in single-radio WMNs and presents a classification for multi-channel protocols for this type of networks. Protocols for single-radio networks are grouped in Refs. [8–18]. Section 4 lists relevant concerns found in multi-radio networks. Then, a classification for multi-channel protocols and architectures for this kind of networks is proposed. Refs. [19–37] refer to these channel assignment approaches. A summary of channel assignment techniques is presented in Section 5, and open research issues are discussed in Section 6. Finally, the conclusion is given in Section 7.

## 2. Multi-channel wireless mesh networks

In WMNs, mesh routers can have more than one interface, which can be tuned to different channels, forming a multi-channel multi-hop WMN as shown in Fig. 1. Some mesh routers also act as gateways between the mesh networks and external networks such as the Internet. A wireless link is established between two neighboring nodes when they have at least one interface tuned to the same channel. This architecture allows enhanced aggregate network throughput by segregating the collision domains into multiple non-overlapping channels and exploiting co-channel reuse; i.e., simultaneous transmissions can occur if transmitting nodes use non-overlapping channels, or they use overlapping channels but are sufficiently distant from each other.

The 802.11 b/g standards use the 2.4 GHz band. The spectrum is divided into 14 overlapping staggered channels whose center frequencies are five megahertz (MHz) apart [2]. Given the separation among channels 1, 6, and 11, the signal on any of these channels is sufficiently attenuated to minimally interfere with a transmitter on any other channel. The 802.11a standard occupies a section of spectrum known as unlicensed national information infrastructure (U-NII) band. The band takes up 300 MHz of spectrum and is divided into three sections of 100 MHz. The first two are next to each other, and the third is 375 MHz up from the top of

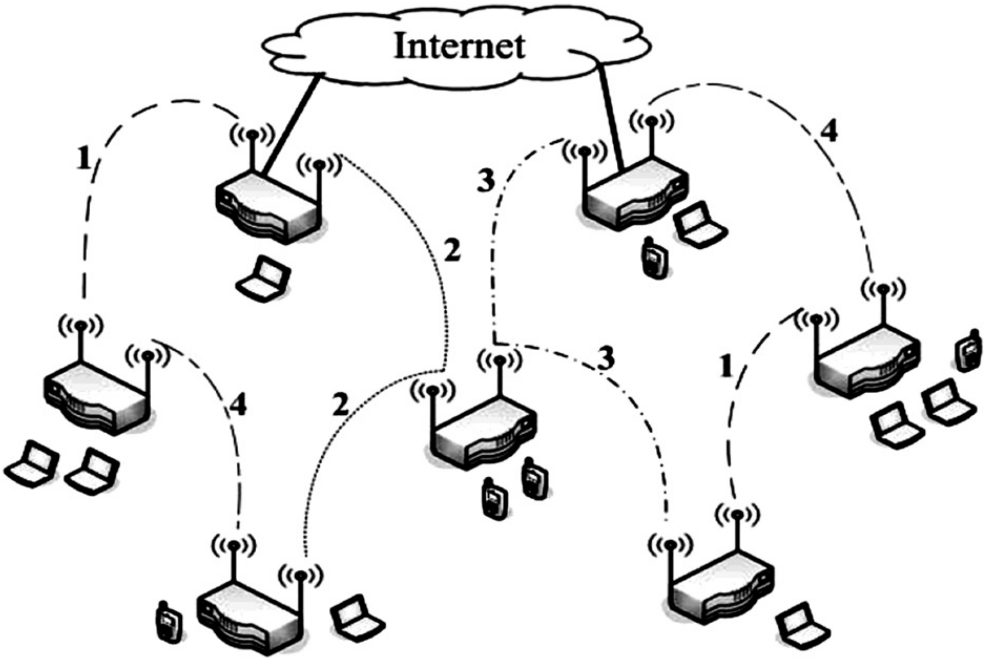

Fig. 1. A multi-channel multi-radio wireless mesh network [23].

the second band. The low band runs from 5.15 GHz to 5.25 GHz, the middle band runs from 5.25 GHz to 5.35 GHz, and the high band runs from 5.725 GHz to 5.825 GHz. Due to the separation among channels, 12 of them can be considered as non-overlapping channels.

IEEE 802.11-based WMNs can be modeled as a collection of nodes equipped with one or more half-duplex, omni-directional radios, where each radio has a transmission range $R$. The assumption of equal transmission range of every radio is for simplicity; if needed it can be easily extended to different transmission ranges as proposed in [4]. A pair of nodes has a link between them in the *undirected connectivity graph* if the distance between them is less than or equal to $R$; i.e., they are located within each other's transmission range. Fig. 2b shows the connectivity graph of the network in Fig. 2a, which depicts four nodes and their transmission ranges.

Let $C_1, C_2, \ldots, C_K$ be the $K$ available channels, and $I_i$ be the number of interfaces at node $i$. There exists a link tuned to $C_q$, $1 \leqslant q \leqslant K$, between a pair of nodes in the *network topology* if there is a link between them in the connectivity graph and they have at least one interface tuned to $C_q$. Note that the assignment of channels to the interfaces induces the network topology. Fig. 3a and b show two different network topologies for the wireless network of Fig. 2a, where $I_A = I_C = 1$, and $I_B = I_D = 2$.

Different channel assignment schemes exist for single-radio and multi-radio networks. Approaches for single-radio networks mostly propose innovative MAC protocols that rely on co-ordination mechanisms to dynamically negotiate channels before each data exchange. They try to optimize the usage of the channels while avoiding new type of problems pertaining to multi-channel environments, namely, the *deafness*, *broadcast*, *deadlock* and *multi-channel hidden terminal problems* [15,33]. On the other hand, multi-radio networks can ease the co-ordination among nodes, since the interfaces may be *statically* tuned (or *semi-dynamically*, at a slow time scale – minutes or hours) and still concurrently used. However, channel assignment strategies should consider issues such as *network partition*, *channel dependency* and *topology alteration* that may occur in multi-radio networks.

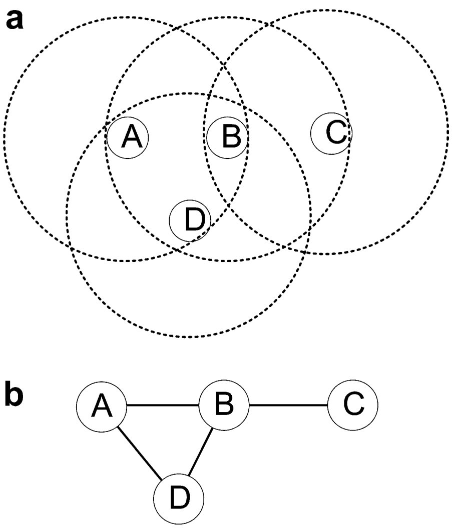

Fig. 2. (a) Wireless nodes and their respective transmission ranges. and (b) undirected Connectivity graph.

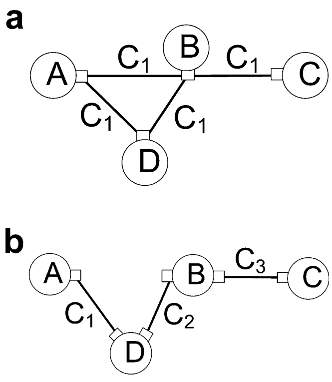

Fig. 3. Two network topologies induced by different channel assignments.

From the fact that multi-channel protocols and architectures for single-radio and multi-radio networks substantially differ from each other, we discuss them separately in the next sections.

## 3. Multi-channel single-radio networks

The channel co-ordination and selection mechanisms play main roles in protocols for multi-channel single-radio networks. The co-ordination mechanism should not only provide suitable ways to negotiate channels but also avoid new problems that result from the use of multiple channels. These problems along with channel selection mechanisms are presented in Section 3.1. Then, the taxonomy is given in Section 3.2.

### 3.1. Multi-channel single-radio issues

#### 3.1.1. Concerns using multiple channels

In order to bring up some concerns which can impact on the performance of multi-channel protocols, we describe a simple multi-channel protocol as an example [33]. Assume that there are three available channels, $C_1$ to $C_3$. The protocol statically assigns a channel to each node of a four-node network, say, $C_1$ to node $A$, $C_2$ to node $B$, $C_3$ to node $C$ and $C_2$ to node $D$. When a node is idle, it stays on its own channel listening for any potential sender; on receiving an RTS packet, it replies with a CTS packet and waits for the data transfer on the same channel. When a node has data to send, it tunes its interface to the receiver's channel, sends an RTS packet and waits for the CTS packet; on receiving the CTS packet, it starts transmitting the data using the receiver's channel. After finishing the data transfer, the sender tunes the interface back to its own channel.

- Multi-channel hidden terminal problem
  Assume that node $A$ has data to transfer to node $B$ in the scenario shown in Fig. 4a. Node $A$ starts by tuning its interface to $C_2$, the channel assigned to node $B$, and sends the $\mathrm{RTS}_1$ packet. Node $B$ then replies with the $\mathrm{CTS}_1$ back to node $A$. After this packet exchange, the data transfer from $A$ to $B$ starts on $C_2$. Assuming that node $C$ was listening to its channel, $C_3$, when the previous RTS/CTS exchange happened, it is unaware of the data transfer between nodes $A$ and $B$ on channel $C_2$. Therefore, node $C$ might initiate a communication with $D$ by sending $\mathrm{RTS}_2$ on $D$'s channel $C_2$, leading to packet collisions at node $B$. This problem is known as multi-channel hidden terminal problem. Note that it is similar to the original hidden terminal problem. However, in a single-channel network, the $\mathrm{RTS}_2$ sent by node $C$ would be prohibited by the earlier $\mathrm{CTS}_1$ packet. Unfortunately, in a multi-channel environment, RTS and CTS packets sent on a given channel cannot be heard by other nodes tuned to different channels. Thus, the channel state information in the form of network allocation vector (NAV) cannot be created [33].

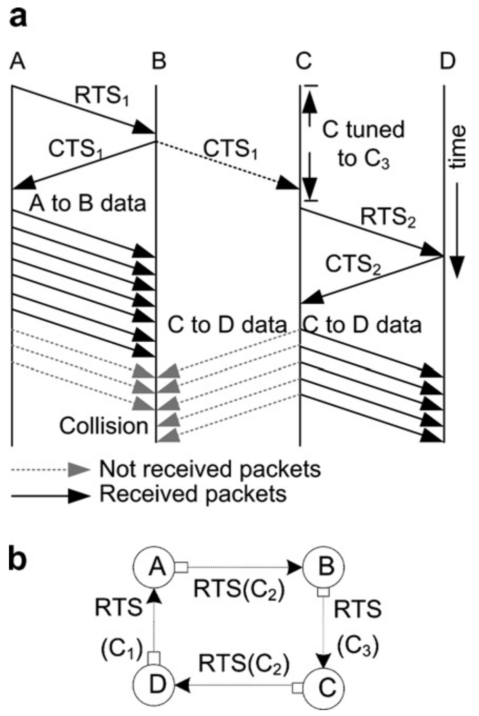

Fig. 4. (a) Multi-channel hidden terminal problem and (b) channel deadlock problem.

- Deafness problem
  Consider again the network of Fig. 4a, but assume that at start time node $B$ is exchanging data with node $C$ on $C$'s channel $C_3$. Node $A$ then attempts to initiate the data transmission to node $B$ by sending $\mathrm{RTS}_1$ on $B$'s channel $C_2$. Unfortunately, that packet is lost since node $B$ has tuned its interface to channel $C_3$. Furthermore, since node $A$ cannot sense the carrier from node $B$ (currently on channel $C_3$), its multiple RTS retrials are lost until it gives up. Node $A$ then falsely concludes that node $B$ is unreachable. This problem is known as deafness problem, and it arises when the intended receiver cannot receive the RTS packets addressed to it on a given channel because its interface is tuned to a different channel [33].
- Channel deadlock problem
  Consider Fig. 4b, where node $A$ attempts to communicate with node $B$ on $B$'s channel $C_2$, which in turn attempts to send an RTS to node $C$ on $C$'s channel $C_3$. A circular dependency is formed since node $C$ is trying to communicate with node $D$, which in turn wants to exchange data with node $A$. Although the deadlock may be resolved automatically, it may degrade the performance of the system [33].
- Broadcast problem
  The broadcast problem results from the fact that some higher layer protocols rely on the broadcast support at the MAC layer; e.g., routing protocols such as DSR [5], AODV [6] and MR-LQSR [28]. Given that multi-channel protocols allow nodes to be tuned to different channels, broadcast packets transmitted in any channel are likely to reach only some of the nodes within the physical communication range.

### 3.1.2. Channel selection mechanisms for single-radio protocols

In single-radio networks, every data transmission is preceded by a channel selection process. There are two basic methods to evaluate the state of a channel: the measurement-based and the status-based [9]. In the measured-based method, a node periodically measures the signal strength and the signal-to-noise ratio (SNR) of each channel to know its status. In the status-based method, each node acquires the channels' busy/idle status by listening to the MAC-layer control packets. Some channel selection mechanisms belonging to one of these two methods are discussed below.

- Clearest channel at the transmitter
  This technique requires the sender node measure the level of carrier signals on all the channels, which can be classified as idle or busy. An idle channel is the one where the total received signal strength is below a sensing threshold. The sender then selects the idle channel with minimum carrier power. Note that no control channel is necessary. Therefore, all the channels can be used for data traffic. The technique can be improved by using *soft reservation* [8], which attempts to assign the same channel that the sender node has used before; if the most recently used channel is idle, then the sender selects that channel; otherwise, it can choose the idle channel with minimum received signal strength. Soft reservation tends to reserve a channel for each node in order to minimize the chance of channel contention.The main drawback of this technique is that the channel is selected by considering signal powers at the sender only, which may result in the selection of a busy channel at the receiver.
- Channel-list
  Channel-list techniques require a control handshake before any data exchange in order to allow the sender to know the condition of all the channels at the receiver side. Then, the sender can choose the most suitable channel based on this information. Channel-list techniques are usually receiver-based; i.e., they attempt to select the *best* channel at the receiver side. The sender and receiver sense the carrier on all the data channels and build a list of channels available for transmission in their respective range. The set of available channels is known as *Preferable Channel List* (PCL). The PCLs are then exchanged by embedding them into control packets. Channel-list techniques can be classified into: (i) based on source–destination counter; (ii) based on packet counter; and (iii) based on power sensing.
- Channel selection based on source–destination counter
  With this technique, nodes keep track of the state of the channels by using a counter for each channel. A channel counter stores the number of source–destination pairs that plan to use or are using the corresponding channel to exchange data. The PCL is then composed of the set of counters. When node $A$ wants to send data to node $B$, it sends its PCL attached to the RTS to node $B$, which selects a channel and notifies back to node $A$ by including the selected channel in the CTS packet. Node $B$ selects the *best* channel based on the PCLs of both sender and receiver; if there is a free channel (i.e., its corresponding counter is equal to zero) in both PCLs, that channel is selected. Otherwise, different options can be considered. For example, a channel that is free at one side can be chosen. On the other hand, if there is no free channel at any side, the channel with the lowest counter can be selected. The CTS packet is also used to announce the selected channel to the neighbors of node $B$. Thus, these nodes can increment their channel counter accordingly. Similarly, the protocol can also include a control packet sent by node $A$ to announce the selected channel to its neighbors [15].
- Channel selection based on packet counter
  The previous technique assumes that every source–destination pair will deliver the same amount of traffic, which is generally not true. Another option is to count the number of scheduled packets to be transmitted through each channel. This approach is similar to the source–destination counter, except that the counters now store the number of packets to be transmitted through the corresponding channels [15].
- Channel selection based on power sensing
  This technique requires to both, sender and receiver, sense the carrier on all the data channels available for transmission. Similar to the clearest channel at the transmitter technique, a channel is classified as idle or busy. The total received signal strength of an idle channel is also stored in order to break ties. A PCL is composed of the idle channels as well as their total received signal

strength, and is embedded into control packets. The receiver node selects the best channel based on both PCLs, similarly to the two previous selection mechanisms [8].

### 3.2. *Classification of channel assignment protocols for single-radio networks*

Considering the operation principle of the coordination mechanisms, multi-channel protocols for single-radio networks can be classified into *Dedicated Control Channel* [8,9], *Hopping* [10–13], *Split Phase* [14–17] and *Receiver-fixed* [14,18]. This classification is an extension of the classification proposed by Mo et al. [7].

#### 3.2.1. *Dedicated control channel*

The idea of the dedicated control channel protocols [8,9] is to isolate control packets from data by dedicating a fixed channel to exchange RTS and CTS packets, and to avoid the interference between control and data packets; the remaining channels are intended for data traffic only. Before starting to send data, a sender node *A* sends an RTS on the control channel to its intended receiver, node *B*. The RTS contains a list of potential channels to be used during the data transfer. Node *B* replies with a CTS, which includes the channel selected for data transfer. Both packets also include the NAV to inform other nodes the duration of the transfer. Nodes *A* and *B* then tune their respective interface to the selected channel to start exchanging data. Fig. 5 illustrates the operation of a dedicated control channel protocol.

Dedicated control channel protocols are presented in [8,9]. In [8], Jain et al. proposed a protocol called Multiple Channel Carrier Sense Multiple Access (M-CSMA). M-CSMA uses the channel-list based on power sensing as its channel selection mechanism. In [9], Li et al. proposed a similar protocol that mainly differs from [8] in the channel selection mechanism. To build its status-based PCL, each node stores in a local table the state of each channel (busy/idle), including the nodes currently using the channels and the time they will be released. A channel-list receiver-based negotiation then takes place on the control channel.

A key issue in the design of a dedicated channel protocol is how to select channels. The selection mechanisms based on source–destination counter and packet counter may not be suitable, since the nodes may suffer from the multi-channel hidden terminal problem, which in turn makes it difficult for them to keep the counters up-to-date. Therefore, the channel selection based on power sensing may be more appropriate.

*The pros of the dedicated control channel approaches*: they are usually easy to implement. There are no additional control packets introduced besides the regular 802.11 RTS/CTS/ACK packets. Moreover, since data and control packets are transmitted on different channels, the requirement of strict synchronization is relieved; thus, the implementation cost is reduced as well.

*The cons of the dedicated control channel approaches*: they mainly suffer from the deafness and multi-channel hidden terminal problems. In addition, in order to build the PCL, a node needs to simultaneously sense carriers on all the channels, which implies extra processing and (potentially) complex decision making to choose the best channel. Scalability may be an issue if the number of nodes in the network keeps increasing, since the control channel can become a bottleneck. In addition, cost efficiency is another issue: in the 2.4 GHz band, there are only three orthogonal channels. Thus, 33% of the channel resource is consumed exclusively for control purposes. Although the authors in [8] assume that the bandwidth of the control channel can be set as needed, this solution is not achievable in 802.11-based networks,

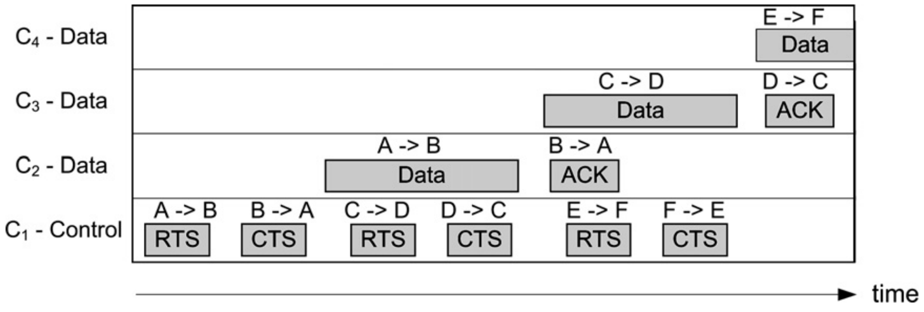

Fig. 5. Operation of a dedicated control channel protocol.

where channels have the same bandwidth. Moreover, even under that assumption, how to compute an appropriate value for the control channel bandwidth is not trivial: a wide control channel may waste bandwidth, while a narrow one may result in congestion. The broadcast support is not guaranteed. Although using the control channel for this purpose assures that broadcast packets are received by idle nodes [8,9], other nodes involved in data exchanges have their interface tuned to different channels.

### 3.2.2. Hopping

In hopping protocols the nodes hop (switch) across multiple narrow-band frequencies. The hopping patterns can be common for all the nodes [10,11] or can be different [12,13].

- Common hopping
  In the common hopping technique [10,11], all the nodes listen to the same frequency at the same time, unless instructed otherwise. Nodes wanting to exchange data then carry out a handshake to remain in the current channel, while the rest of the nodes continue hopping on the common sequence. The basic operation of this technique is shown in Fig. 6. Nodes follow the hopping sequence $C_1$, $C_2$, $C_3$. The time is slotted in discrete intervals given by the dwell time, which should be at least as long as the time needed to receive a control packet. At time $t_1$, the system is at hop $C_1$, and node $A$ sends an RTS to node $B$. At time $t_2$, all other nodes except $A$ and $B$, which replies with the CTS back to $A$, hop to $C_2$. Simultaneously at $t_2$, node $C$ sends an RTS on channel $C_2$ to node $D$, which replies at $t_3$ with the corresponding CTS. By time $t_4$, two simultaneous transmissions are taking place at $C_1$ and $C_2$, respectively.Common hopping protocols are presented in [10,11]. Tzamaloukas et al. [10] proposed a protocol called Channel-hopping Multiple Access (CHMA). CHMA works without carrier sensing: a node attempting to transmit simply initiates the RTS/CTS handshake. The same authors presented Channel Hopping Access with Trains (CHAT) [11]. CHAT uses a more complex handshake that requires changes to the 802.11 standard to allow collision-free transmissions of packet trains and to implement multicast transmissions. A packet train consists of two or more data packets that can be addressed to different destination nodes.
- Independent hopping
  Independent hopping protocols [12,13] also divide the time in discrete intervals or slots. Nodes do not share a common hopping sequence. Instead, they switch to (probably) different channels according to their individual sequence. Nodes then iterate through their own hopping sequence, staying in a channel for a dwell time. Network partitions may be prevented by requiring all nodes hop to a predetermined channel after they have iterated through every channel of their own sequence. Nodes, then, overlap at least during one slot per sequence cycle, which permits them to exchange and learn each other's hopping sequence. When a sender wants to transmit data, it hops to the next channel of the receiver's hopping sequence. There are two variants at this point. In the first variant, the data transfer is completely performed in that channel. An example of this kind of protocol is McMac [13]. McMac does not implement the predetermined hop intended to avoid network partitions, but simply relies on the fact that nodes will overlap during their hopping sequence. Each node announces its hopping sequence by attaching its seed (used to compute the hopping sequence) into every packet. Nodes then eventually learn each other's hopping sequences. In the second variant, the data transfer is performed over successive hops of the receiver's sequence.

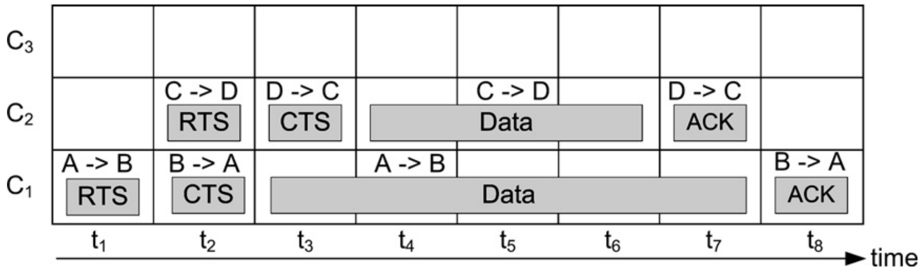

Fig. 6. Operation of a common hopping protocol.

Slotted Seeded Channel Hopping (SSCH) [12], which is illustrated in Fig. 7, is an example of the second variant. Assume that nodes *A* and *B* have the following respective hopping sequences: $C_4$, $C_1$, $C_2$ and $C_2$, $C_4$, $C_1$. Their first hopping sequence cycle finishes at time $t_3$; at time $t_4$ they hop to $C_3$, the predetermined channel intended to avoid network partitions. From time $t_6$, node *B* follows *A*'s hopping sequence to send data to it, until the eventual completion of the data transfer. SSCH is implemented on top of 802.11 MAC layer and does not introduce extra overhead to the RTS/CTS mechanism.

*The pros of the channel hopping approaches*: they eliminate the potential channel bottleneck problem. In addition, the decision making process before a data exchange is trivial; a sender simply follows either the common sequence (common hopping protocols) or the receiver's sequence (independent hopping protocols). These protocols neither introduce extra overhead to the RTS/CTS mechanism nor require channel negotiations before a data transmission.

*The cons of the channel hopping approaches*: they require supports from the synchronization mechanisms, especially the common hopping protocols, where nodes must be tightly synchronized. In addition, the channel switching delay may significantly degrade the performance. Current RF transceivers [38,39] show a typical channel switching delay of about 150 μs. Thus, hopping protocols must set the dwell time properly such that the switching overhead is amortized. In [12], the authors measured a switching and synchronizing overhead of 3.3% in SSCH. Channel hopping protocols may also suffer from the deafness and the multi-channel hidden terminal problems. Particular issues observed in common hopping protocols [10,11] include: (i) they are not back compatible with the 802.11 standard, and (ii) in order to evaluate the behavior of the non-carrier sense mechanism in extreme conditions, a deeper quantitative analysis under saturation cases (source nodes always have packets to transmit) would be desirable. Similarly, issues observed in independent hopping protocols [12,13] include: (i) the synchronization mechanism that guarantees that any two neighboring nodes eventually overlap is non-trivial to develop, especially in networks with a large number of nodes. Although an independent hopping protocol may work well in an *optimistic* case, where a sender knows the receiver's hopping sequence, it might incur in high latency penalties if the sender does not have information about the receiver or if the information is out of date; (ii) a node must keep track of the hopping sequence of its neighbors. In a dense network, this may be limited by the amount of available memory; and (iii) the broadcast support is not guaranteed due to similar reasons exposed for dedicated control channel protocols. The problem might be alleviated by retransmitting broadcast packets over a large number of slots [12]. However, this solution does not guarantee that the packets will be received by each neighbor. Moreover, the retransmission over many slots may not be compatible with all uses of broadcast, such as a broadcast application for synchronization purposes [40]. Finally, the number of retransmissions should be network-size dependent, which makes this solution not general.

### 3.2.3. Split phase

In the split phase protocols [14–17], the time is divided into cycles composed of two phases: control phase and data exchange phase. Nodes are assumed to be synchronized. At the beginning of each interval, the control phase takes place and all the nodes listen to a common channel. This phase is intended to negotiate a channel to be used during the data exchange phase. When the control phase ends, the data exchange phase starts. Fig. 8 illustrates the operation of a split phase protocol. Nodes *A* and *C* want to transmit data to nodes *B* and *D*, respectively. When a new cycle starts, every node switches to $C_1$, the control channel. Since node *A* has data to send, it waits a random time (to avoid collisions)

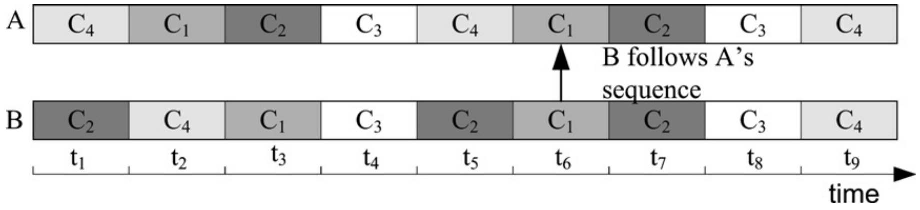

Fig. 7. An example of an independent channel hopping protocol.

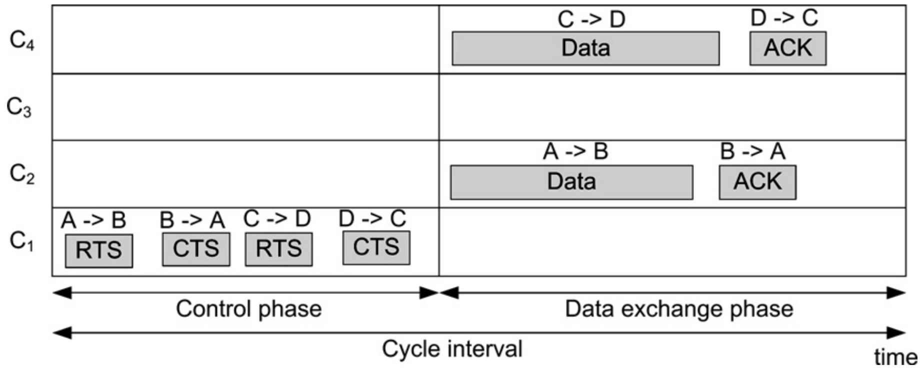

Fig. 8. Channel negotiation and data exchange in a split phase protocol.

and sends an RTS[4] including its PCL to node *B*, which sends a CTS back to *A* with its preferred channel or PCL. After this handshake, nodes *A* and *B* agree to use a given channel, say $C_2$, during the next data exchange phase. From the RTS and CTS packets, the neighbors of *A* and *B*, namely *C* and D, know that $C_2$ will be busy during the next data exchange phase. Therefore, when node *C* sends an RTS to node *D*, it does not include $C_2$ as a preferred channel; instead, it picks up an available channel, say $C_4$. The channel negotiation may include a final control packet from the sender to announce to its neighbors the channel selected for the next data exchange phase.

An important issue in split phase protocols is the number of negotiations allowed during the control phase: if such a number is limited by *K*, the number of channels, then each source–destination pair granted with a channel has fully access to it during the entire data exchange phase. This scheme was proposed by Maheshwari et al. [14], who developed a protocol called Local Co-ordination-based Multi-channel MAC (LCM). LCM dynamically adjusts the duration of both the control and data exchange phases. The protocol implements a status-based channel selection mechanism. A less restrictive protocol may allow more than *K* channel negotiations during the control phase. However, a channel access mechanism may then be needed to avoid or resolve contentions among pairs of source–destination nodes choosing the same channel. Depending on the channel access mechanism, split phase protocols can be classified into *carrier-sense-based* [15] and *TDMA-based* [16,17] (TDMA stands for time division multiple access).

- Carrier-sense-based
  This scheme enforces each source–destination pair to exchange RTS/CTS packets again in the data exchange phase. Then, only the winner pair acquires the channel. An example of carrier-sense-based protocol called Multi-Channel MAC (MMAC) was presented in [15]. MMAC sets the duration of both control and data exchange phases to a fixed value. The channel selection mechanism is based on the source–destination counter technique.
- TDMA-based
  A TDMA-based protocol called ADHOC MAC for single-radio networks was proposed by Borgonovo et al. [43]. Although its use in multi-channel environments was not discussed, some relevant features applicable to multi-channel protocols include the use of a time slot structure, the reservation of slots to achieve a contention free interval, the potential provision of different levels of Quality-of-Service (QoS) and the broadcast service. The latter constitutes a main advantage over the protocols discussed previously. In TDMA-based multi-channel protocols, contentions during the data exchange phase are mitigated by making source–destination nodes negotiate not only the channel but also the initial time and duration of each data transmission. In [16], Chen et al. presented Multi-channel Access Protocol (MAP). In MAP, the duration of the control phase is fixed while the duration of the data exchange phase depends on the scheduling established during the control phase. Having listened to every negotiation during the control phase, a sender node performs a scheduling

---

[4] The packets sent in the control phase need not be RTS/CTS. Other suitable packets may be used.

algorithm that determines the channel and the initial time for each transmission. The scheduling algorithm guarantees that any two source–destination pairs do not overlap over time on any channel. In [17], Zhang et al. presented TMMAC. Two main differences can be noted between MAP and TMMAC: (i) in TMMAC, the time during the data exchange phase is slotted, as in ADHOC MAC. The duration of each slot is defined as the time needed to transfer a single data packet, including the channel switching delay and the transmission of both data and ACK packets; and (ii) while MAP applies a scheduling algorithm after having listened to every channel negotiation, in TMMAC a source–destination pair reserves time slots as needed during the control phase. TMMAC sets the cycle interval to a fixed value while allowing a dynamic adjustment of the control phase at the expense of the data exchange phase duration.The problem of finding a contention free assignment on the available channels is not a new concern. Schemes for infrastructure networks were proposed in [56], where issues such as energy consumption and QoS were addressed. Such scheduling algorithms, though not originally designed for multi-hop networks, may provide insight into TDMA-based protocol designs.

How to set the duration of both control and data exchange phases is a challenging issue in split phase protocols. If the number of source–destination pairs is small, the control phase may be longer than the necessary. A large portion of the control phase might be left as idle because data packets can not be transmitted in that interval. On the other hand, if there are many source–destination pairs, a longer control phase may be needed. Similarly, if the data exchange phase is too short, the time overhead of the control phase may not be amortized. On the contrary, if the data exchange phase is too large, a significant amount of bandwidth may be wasted by nodes that run out of packets to send. In addition, nodes waiting for the next control phase may experience an excessive delay. Table 1 summarizes the approach of each protocol previously discussed.

Table 1
Duration of the control phase, the data exchange phase and the cycle interval

| Phase | LCM [14] | MMAC [15] | MAP [16] | TMMAC [17] |
|---|---|---|---|---|
| Control | Variable | Fixed | Fixed | Variable |
| Data exchange | Variable | Fixed | Variable | Variable[a] |
| Cycle interval | Variable | Fixed | Variable | Fixed |

[a] The duration of the data exchange is equal to the cycle interval minus the control phase duration.

*The pros of the split phase approaches*: they mitigate the deafness and multi-channel hidden terminal problems by requiring every node listen synchronously to the control channel during the control phase. In addition, they eliminate the channel bottleneck problem and utilize all the channels during the data exchange phase. This contrasts with the operation of dedicated control channel protocols, where the control channel cannot be used to exchange data. They also provide solutions to the broadcast problem. In LCM [14] and MAP [16], the authors suggested supporting broadcast services by requiring a sender node send broadcast packets during the control phase. In TMMAC [17], although a node trying the send a broadcast packet contends during the control phase with nodes trying to send unicast packets, the node is given higher priority by reducing the size of its MAC backoff window (used during the control phase), which increments its probability of accessing the medium earlier to reserve time slots.

*The cons of the split phase approaches*: issues observed in these protocols include: (i) they rely on tight synchronization mechanisms; (ii) it may be complex to compute the proper duration of both control and data exchange phases; (iii) the channel switching delay affects the performance of the protocol. Although the switching delay can be amortized by increasing the data exchange phase duration, this solution can also have counter effects, as explained before; (iv) extra processing is needed to simultaneously sense carrier on all the channels; (v) during the control phase, all the channels except the one used for control purposes remain idle. In order to alleviate this problem, Chen et al. [16] suggested not enforcing stations stay tuned to the control channel during the control phase. However, this can lead the protocol to suffer from the broadcast, deafness and multi-channel hidden terminal problems; and (vi) split phase protocols proposed so far require changes in the 802.11 MAC layer. MMAC [15] shows additional drawbacks, namely: (i) during the data exchange phase, each pair of source–destination nodes needs to apply the RTS/CTS mechanism. Thus, two negotiations are needed to perform a data exchange (the previous negotiation takes place during the control phase), which increases the total overhead; and (ii) the control and data exchange phase durations are fixed. Similarly, an issue found in MAP [16] is that, in order to apply the proposed scheduling algorithm, a node has to listen to all the negotiations. If a node misses any negotiation, its computed schedule is likely to be inconsistent with other's schedule, which can result in collisions during the data exchange phase. In TMMAC [17], it can be observed that: (i) the time slot depends on the packet size. Moreover, TMMAC assumes that the interfaces operate at a predefined data rate (e.g., 2 Mbps is set for simulation purposes), which may not fully exploit the capacity of the radios; (ii) in case of more than one time slot on different channel are available, the channel selection scheme does not consider a mechanism to break ties. A suitable mechanism may consider the link quality of each channel; and (iii) the assumption of 80 μs as a switching delay does not hold for many transceivers [38,39].

### 3.2.4. *Receiver-fixed*

Receiver-fixed protocols [14,18] assign a fixed *quiescent* channel to each node. To send data, a node tunes its interface to the quiescent channel of the intended receiver, which, when idle, listens to that channel. The data transfer is then performed in the quiescent channel of the receiver. Following a successful transfer, the sender tunes its interface back to its own quiescent channel. Fig. 9 illustrates the operation of a receiver-fixed protocol. Assume that the quiescent channels of nodes $A$, $B$ and $C$ are $C_1$, $C_2$ and $C_3$, respectively. At time $t_1$, node $C$ has data to send to node $B$; therefore, it tunes its interface to $B$'s quiescent channels, $C_2$, and sends an RTS. After having received the corresponding CTS back, node C starts the data transmission. Finally, when the ACK from node $B$ is received at $t_2$, node $C$ switches its interface back to $C_3$.

Examples of receiver-fixed protocols are presented in [14,18]. Shacham et al. [18] proposed the Receiver Directed Transmission (RDT) protocol. RDT assumes that the assignment of the quiescent channels as well as the mechanism that allows nodes to learn each other's quiescent channel are separately performed (e.g., this function might be implemented by an upper layer as in SSCH [12], which would make RDT compatible with the 802.11 standard). Maheshwari et al. [14] extended RDT (xRDT) by incorporating a *tone interface* to each node and a *busy tone* for each data channel. Busy tones are single frequency tones intended for signaling. When a node starts receiving data on a channel, say $C_1$, it turns the corresponding busy tone on. The busy tone notifies any potential transmitter newly arrived at $C_1$ to defer its transmission on that channel, avoiding the multi-channel hidden terminal problem. The protocol also incorporates a messaging mechanism that alleviates the deafness problem.

*The pros of the receiver-fixed approaches*: before a data exchange, receiver-fixed protocols only need to perform a very simple decision making, which can be easily implemented in software over the 802.11-compliant cards.

*The cons of the receiver-fixed approaches*: broadcast support is not discussed in [14,18]. One way to offer such service is by broadcasting in every channel. The drawback of this solution is the high bandwidth consumption. The deafness and multi-channel hidden terminal problems strongly impact on the performance of these protocols (for a detailed explanation, refer to Section 3.1.1, where a receiver-fixed protocol is used as an example to explain these problems). Simulation results [14] show that the deafness problem occurs frequently in a heavily loaded scenario, so that a receiver-fixed protocol (RDT) may perform even poorer than the 802.11 MAC single-channel protocol. To solve the multi-channel hidden terminal problem, Maheshwari et al. [14] suggested the incorporation of the

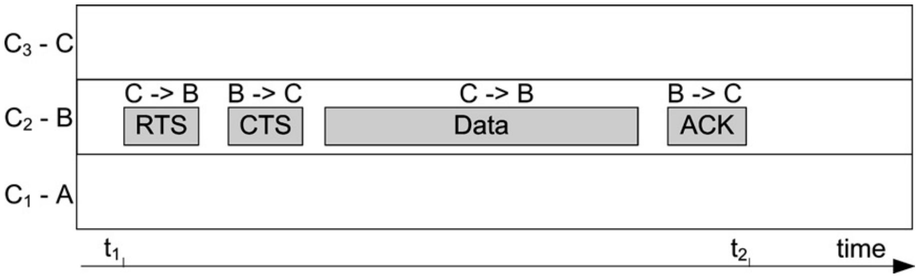

Fig. 9. Data exchange in a receiver-fixed protocol.

tone interface. However, adding such special-purpose hardware may not be the best option. Instead, another 802.11 card can solve the problem and, at the same time, increase the capacity of the network (multi-radio). Due to the low cost of 802.11 radios, this can be a preferred option.

## 4. Multi-channel multi-radio networks

Multi-radio approaches, unlike single-radio protocols, need not implement a co-ordination mechanism to achieve concurrent transmissions. For example, in Fig. 3b, note that the three links can operate simultaneously. In addition, nodes *B* and *D* can send and receive data at the same time. In general, channels may be statically or semi-dynamically assigned in order to avoid dynamic channel negotiations that require switch the radios between successive transmissions. However, new potential problems arise when semi-dynamic reassignments are allowed, which are discussed, along with the *protocol model of interference* [41], in Section 4.1. Then, a classification for channel assignment approaches for multi-radio WMNs is given in Section 4.2.

### *4.1. Multi-channel multi-radio network issues*

### *4.1.1. Concerns using multiple channels*

In Section 3.1.1, we enumerated some concerns found while designing single-radio protocols, namely broadcast support, deafness, channel deadlock and multi-channel hidden terminal problems. In multi-radio networks, such concerns are mitigated by equipping nodes with more than one interface and by assigning them channels in a *relative* static way. To illustrate this idea, consider again the network shown in Fig. 2a. In a single-radio network, if node *D* wants to send data to node *C*, two channel negotiations are needed: one for hop *D*–*B* and another for hop *B*–*C*. On the other hand, in the multi-radio network shown in Fig. 3b, node *D* simply accesses $C_2$ to send data to node *B*, which in turn accesses $C_3$ to forward the packets to node *C*; i.e., no dynamic channel negotiations are needed. Note that the deafness and multi-channel hidden terminal problems are mitigated. Recall that the deafness problem arises because an intended *one-hop receiver*, node *B* in the above example, may currently be in a different channel from its intended transmitter, node *D*. This situation is avoided in Fig. 3b, since nodes *B* and *D* have one interface tuned to $C_2$. Similarly, recall that the multi-channel hidden terminal problem occurs because nodes may listen to different channels while negotiations in the form of RTS/CTS take place, which makes it difficult to use virtual carrier sensing to avoid the hidden terminal problem. Since the interfaces are more statically tuned in multi-radio networks, nodes are listening to the same channel for longer periods of time. Thus, they are able to receive RTS/CTS packets on the assigned channel and properly set the NAV. The broadcast problem is handled by broadcasting packets through every interface. Note that a node receiving a broadcast packet may need to also forward through the interface from which the packet was received, in case of the node is an intermediate hop on that channel. Assuming a connected network, a broadcast packet will eventually be received by every node.

In multi-radio networks, channel reassignments, in a lower time scale, may be desirable for reasons that include: to minimize the interference from external networks; to improve the capacity when some links are heavily used; and to balance the load over different channels. They, nevertheless, introduce new concerns that are discussed below.

- Network partition problem
  Consider the Fig. 10a and assume that the interface of node *C* tuned to $C_1$ is reassigned to $C_3$ as shown in Fig. 10b. If nodes *D* and *E* do not co-

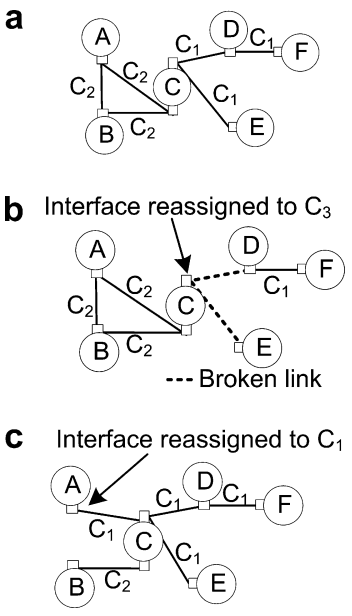

Fig. 10. (a) A multi-radio network. (b)–(c) Channel reassignments at nodes *C* and *A*, respectively.

ordinate with node $C$ during the reassignment, the network becomes partitioned, since links $(C,D)$ and $(C,E)$ are broken.

- Channel dependency problem
  From the above example, we note that nodes $D$ and $E$ must reassign their respective interface. Moreover, as a ripple effect, link $(D,F)$ will require node $F$ reassign its interface. This chain effect is known as *channel dependency problem*. The channel reassignment thus needs to be propagated in a co-ordinated manner in order to avoid network partitions.
- Topology alteration and impact on routing
  Channel reassignments may alter the network topology; some links may be created while others may disappear. Coming back to Fig. 10a, if node A reassigns its interface to channel $C_1$ as shown in Fig. 10c, links $(A,B)$ and $(A,C)$ on channel $C_2$ are disrupted, while a new link $(A,C)$ on channel $C_1$ is created. These changes in the network topology can impact on upper layers, especially on routing protocols. If there are flows flowing through links that no longer exist after a reassignment, then they are disrupted. The routing module thus needs to invalidate the affected paths. In addition, new route discovery processes could be needed. This process may take an unacceptable time if the routing protocol is unaware of the channel reassignment, and thus may have chain effects in upper layer protocols. Channel reassignments may also impact on the capacity of new paths. In Fig. 10c, after the reassignment at node $A$, the new path from node $A$ to node $D$ is composed of the links $(A,C)$ and $(C,D)$, both on channel $C_1$, and thus contending with each other. The data rate of the path may then decrease.
- Non-convergent behavior
  How to guarantee that a channel assignment approach leads to a stable network topology is another important issue, especially for a distributed scheme, where a channel reassignment may be triggered at some node that attempts to improve its local conditions. For instance, in the network of Fig. 11a, a multi-channel protocol might reroute the flow $f_{AB}$ from the link on channel $C_1$ to the link on channel $C_2$, as shown in Fig. 11b, in order to eliminate the *inter-flow* interference between $f_{DA}$ and $f_{AB}$. As the result, flows $f_{AB}$ and $f_{BC}$ now contend for channel $C_2$. Node $B$ might then decide to reassign one of its interfaces to channel $C_3$ and to reroute the flow $f_{BC}$ to that channel, as shown in Fig. 11c. Unfortunately, this change affects the flow $f_{CD}$. The channel reassignments might indefinitely continue, leading to a non-convergent behavior. This problem is related with the channel distribution fairness [55], which tries to fairly (all channels are equally exploited) utilize the available channels.

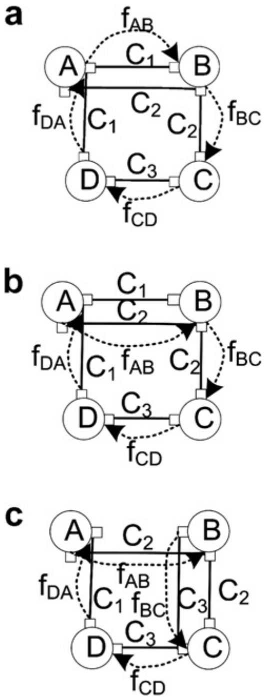

Fig. 11. (a) A four-node network. (b) Flow $f_{AB}$ is rerouted from the link on channel $C_1$ to the link on channel $C_2$. (c) $B$ reassigns an interface to channel $C_3$ and reroutes the flow $f_{BC}$ to the link on that channel.

### 4.1.2. Wireless interference model

Although the wireless interference model presented in this section applies also to single-radio networks, we have delayed its presentation since, in the context of this survey, it is mainly used by multi-radio channel assignment approaches. The existence and extent of interference between a pair of links in a wireless network is determined by an interference model. Gupta and Kumar [41] proposed two models: *protocol model* and *physical model*. The latter is more related to physical layer considerations. We only present the protocol model and its extension for successful transmissions in 802.11 networks. For the physical model, refer to [41].

- Protocol model
  The protocol model [41] determines that a transmission from a node $A$ to a node $B$ is successful if

(i) there exists a link between them in the network topology, which is used for the transmission; and (ii) any node $C$ such that $d_{CB} \leqslant R'$ is not transmitting in the channel used by nodes $A$ and $B$. $d_{CB}$ represents the distance between nodes $C$ and $B$, and $R'$ represents the interference range, which for simplicity is assumed to be the same for all the nodes. Note that this model does not require the sender, node $A$, be free of interference, which is needed for the 802.11 RTS/CTS and Data/ACK packet exchanges. The extended protocol model captures this idea.

- Extended protocol modelA transmission from a node $A$ to a node $B$ is successful if (i) there exists a link between them in the network topology, which is used for the transmission; and (ii) any node $C$ such that $d_{CB} \leqslant R'$ or $d_{\mathrm{CA}} \leqslant R'$ is neither transmitting nor receiving in the channel used by $A$ and $B$.
- Conflict graph

  Conflict graphs, also known as contention graphs, incorporate the wireless interference concepts into the channel assignment problem formulation. They can be defined in terms of traffic flow interference [42] or link interference [4].

  *Flow conflict graph*. The flow conflict graph explained here is defined in terms of the protocol model. Assume that the set of unidirectional traffic flows flowing through each wireless link is known beforehand. A flow $f_{AB}$ from node $A$ to node $B$ is successfully transmitted if the conditions for a successful transmission of the protocol model are satisfied. On the other hand, a flow $f_{CD}$ is said to interfere with flow $f_{AB}$ if $f_{CD}$ takes place on the same channel used by $f_{AB}$ and $d_{CB} \leqslant R'$. The flow conflict graph is defined as a graph $G = (V, E)$, where the set of vertices[5] $V$ corresponds to the set of all flows, and $E$ is the set of edges. A directed edge $(f_{CD}, f_{AB})$ belongs to $E$ if flow $f_{CD}$ interferes with flow $f_{AB}$. Fig. 12 illustrates a network topology with its corresponding flow conflict graph (which assumes that $d_{AC} > R'$). In the flow conflict graph, the edges can be assigned weights to indicate the extent of the interference between flows. Thus, it models the intra-channel interference among flows. Let the flow conflict weight $W_{\mathrm{FAB}}$ be the sum of the weights of the edges incident to the vertex corresponding to the flow $f_{AB}$. Coming back to our example, and assuming a unitary weight on every edge, the flow conflict weights are $W_{\mathrm{FAB}} = 2$; $W_{FBC} = 0$, and $W_{\mathrm{FAD}} = 2$. The flow conflict weights can then be used to define two useful metrics that quantify interference: (i) average flow conflict weight of all the nodes in the conflict graph; and (ii) maximum flow conflict weight [25].

  *Link conflict graph*. Based on the extended protocol model, the link conflict graph is a graph $G = (V, E)$, where $V$ is the set of all wireless links and $E$ is the set of edges. An undirected edge $(AB, CD)$ belongs to $E$ if the second condition of the extended protocol model is not satisfied when simultaneous transmissions through the links $(A, B)$ and $(C, D)$ take place [4]. By assigning weights to the edges, the link conflict weight $W_{AB}$ can be defined as the sum of the weights of the edges incident to the vertex $AB$. Fig. 12c illustrates the link conflict graph of the network in Fig. 12a, where for simplicity links are assumed to be undirected. By considering a unitary weight on every edge, both the maximum and the average link conflict weight are equal to 3.

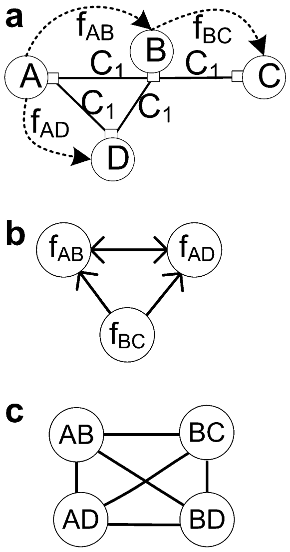

Fig. 12. (a) A network topology with three flows. (b) The corresponding flow conflict graph based on the protocol model. (c) The corresponding link conflict graph based on the extended protocol model.

### 4.2. *Classification of channel assignment protocols and architectures for multi-radio networks*

Based on how frequently the channel assignments are performed, the protocols and architec-

---

[5] As in [4], we associate the terms *node* and *link* with the network topology and connectivity graphs, and the terms *vertex* and *edge* with the conflict graph.

tures for multi-radio networks can be classified as *dynamic* [8–18,31], *semi-dynamic* [19–24], *static* [25–32] and *hybrid* [33–36].

### 4.2.1. Dynamic assignment

Dynamic protocols enforce nodes switch their interfaces dynamically from one channel to another between successive data transmissions. The multi-channel protocols for single-radio networks reviewed in Section 3.2 fall into this category. Although dynamic protocols may be applied to multi-radio WMNs, they may not fully exploit the advantages of multi-radio networks. Thus, in this section we will mainly focus on semi-dynamic, static and hybrid techniques.

### 4.2.2. Semi-dynamic assignment

Semi-dynamic schemes assign or reassign channels at a slow time scale, minutes or hours. According to the criterion used to perform reassignments, they can be classified into *external-interference-aware* [19] and *load-aware* [20–24].

- External-interference-aware

  In addition to consider internal interference (among links of the WMN), an external-interference-aware scheme tries to eliminate or mitigate the interference from external networks by performing channel reassignments. Fig. 13 illustrates the concept of external-interference-aware. Assume that the link of the external network composed of nodes $X$ and $Y$ interferes with link $(C, D)$. By reassigning that link (i.e., switching the corresponding interface at nodes $C$ and $D$) to a different channel, say $C_3$, the external interference is eliminated. Note that nodes $E$ and $F$ must also reassign their respective interface. The internal interference may be quantified by the conflict graph. However, the conflict graph may not capture the external interference, since the network topology of external networks is unlikely available.

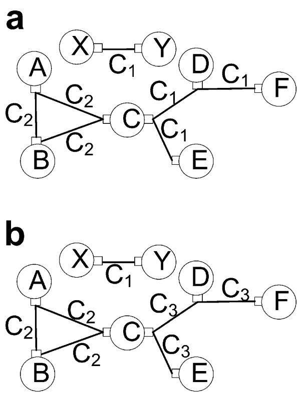

Fig. 13. (a) An external wireless network composed of nodes $X$ and $Y$ interfering with a WMN; (b) after a channel reassignment.

  In [19], Ramachandran et al. proposed and implemented a centralized architecture with 802.11 compliant cards. The central entity quantifies internal interference by constructing an *extended* link conflict graph. To estimate external interference, each node counts the number of *visible* external-interfering radios at its own location. A visible radio is defined as the one whose packets are correctly received. The estimation is periodically sent to the central server, which applies a *Breadth First Search – Channel Assignment* algorithm to perform channel reassignments as needed. The architecture requires all nodes tune one interface to a common default channel, which may be very costly in terms of channel resources.

  *The pros of the external-interference-aware approaches*: they are able to dynamically react to external interference. Due to the explosive grown of the use of the unlicensed bands, this feature can be very useful. To the best of our knowledge, [19] is the only paper that addresses this problem. MeshDynamics [58] also incorporates this feature into its routers. In addition, the architecture proposed in [19] preserves the connectivity between any two neighboring nodes, which avoids network partitions. Moreover, a link redirection protocol is jointly used with the channel reassignment algorithm to avoid flow disruption.

  *The cons of the external-interference-aware approaches*: traffic flows are not considered. A better approach may reassign channels based on both traffic variations and external interference. The reassignments may result in topology alterations, which directly affect the routing module. For example, in [19], WCETT [28] is used as routing metric. Given that WCETT is a *channel-aware* metric, each channel reassignment should interact with the routing module in order to recompute path metrics and, eventually, revalidate paths that might become invalid or not optimal anymore. This tight interaction may be complex to achieve. In addition, it is still a challenging issue how to accurately measure external interference. The procedure proposed in [19] has two drawbacks: (i) external radios within the interference range of the estimating node but outside of its reception range are not taken into account, since packets transmitted by such radios cannot be properly received; and (ii) the amount of traffic generated by interfering radios is not considered. A better approach might measure the number of data packets received at the estimating radio.

- Load-aware

These schemes consider traffic loads and their variations to perform channel reassignments [20–24]. Fig. 14 illustrates an example of a load-aware reassignment. In (a), two flows are transmitted on channel $C_1$, which becomes a shared channel. Based on the per-channel usage, node $B$ can decide to reassign its interface to channel $C_2$ as shown in (b). The reassignment permits the simultaneous transmission of both flows, improving the network throughput.

In [20], Raniwala et al. presented a centralized architecture for channel assignment and routing. Given the node placement and the traffic load between each pair of nodes, the channel assignment algorithm binds each interface to a channel such that the available bandwidth on each link is proportional to its expected load. If the loads change over time, the algorithm can perform channel reassignments. This scheme assumes that the load between neighboring nodes is known before the initial channel assignment, which implies that the set of links is given beforehand. In [21], Raniwala and Chiueh proposed a distributed architecture for routing and channel assignment. To perform reassignments, nodes estimate the usage status of all the channels within their neighborhood. The usage of a channel is proportional to the aggregate traffic load and the number of nodes on the channel. When a node finds a channel with a lower usage, it can perform a reassignment to that channel. The architecture bounds the impact of channel dependencies by imposing a *parent–children* hierarchy among nodes. A node can only reassign an interface if that interface is not used to communicate with its parent. Both architectures presented above are implemented with 802.11 compliant cards. In [22], Alicherry et al. formulated the channel assignment, routing and link scheduling problems jointly. The problem formulation incorporates wireless interference by using the extended protocol model. Given that the formulation results in a mixed integer linear-programming model, they proposed an approximation algorithm to maximize the network throughput under a fairness constraint. The proposed scheme allows channel reassignments when the traffic patterns vary. In [23], Mohsenian et al. formulated the channel assignment problem as a nonlinear program, and proposed an algorithm that computes a *log-quadratic* formula representing the solution of the problem. The objective of the algorithm is the maximization of the *aggregate utility* across all sources. The authors did not implement a protocol supporting the proposed channel assignment algorithm. However, they highlighted the main features of such protocol, for both centralized and distributed architectures. For a centralized architecture, one node solves the nonlinear program and then announces the channel assignments to the rest. For a distributed implementation, each node is responsible for assigning channels to some of its links. In addition, nodes periodically exchange channel usage information in order to locally perform the reassignments. In [24], Wu et al. presented a distributed channel assignment and routing scheme that resides between the MAC and routing layers. The scheme is based on a traffic-dependent metric called Channel Cost Metric (CCM). CCM captures channel utilization and interference in the two-hop neighborhood. A node initiates a (potential) reassignment if the channel utilization of its corresponding interface is higher than a predefined threshold. The reas-

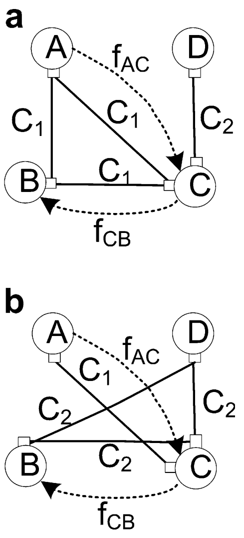

Fig. 14. (a) A network with two flows, $f_{AC}$ and $f_{CB}$. (b) After reassigning the interface of node $B$, $f_{AC}$ and $f_{CB}$ can be concurrently transmitted on different channels.

signment includes a mechanism that assures an improvement in terms of CCM, while preserving connectivity.

*The pros of the load-aware approaches*: channel reassignments due to variation of traffic loads lead to enhanced performance. Raniwala et al. [20] demonstrated by simulations that in a scenario with two interfaces per node, their proposed scheme can yield a factor of up to eight of improvement in terms of throughput with respect to a single-channel network, while a static technique can only achieve a factor of two of improvement. Load-aware approaches can accommodate flows through disjoint orthogonal links and, thus, balance the total load over multiple channels. In [21], the distributed architecture with the parent–children scheme effectively avoids network partitions and predicts the reassignments imposed by channel dependencies while still preserving flexibility. In [23], the channel assignment algorithm considers not only orthogonal channels, but also partially overlapping channels. Simulation results show that the use of such channels may improve the network throughput. In [24], the proposed metric, CCM, permits to quantify the quality of links; i.e., it considers heterogeneous links.

*The cons of the load-aware approaches*: the dynamic traffic profiles may be very challenging to estimate. How to accurately quantify the interference among flows is another key issue. If reassignments are performed in a distributed manner, a heavy protocol overhead resulting from control packet exchanges may be needed to keep the network topology up-to-date. In addition, the network topology may experience a non-convergent behavior as the result of the simultaneous discovery of an underutilized channel by multiple nodes [20]. Moreover, reassignments that only lead to marginal improvement may even negatively impact on the system because of excessive topology alterations; how to provide the best trade-off between performance and stability is still a challenging issue. A particular disadvantage observed in [21] results from the proposed *fat tree* structure. The network topology follows a multi-tree structure, where roots are located at gateways that connect the WMN to the Internet. By assuming a fixed structure, this architecture may have a poor performance when used in an arbitrary WMN. In [20,22], the centralized architectures rely on the fact that traffic patterns are globally known by the central entity. Further issues observed in [22] are: (i) the proposed approximation algorithm may not be practical due to its complexity and assumptions. The complete algorithm for channel assignment, routing and link scheduling performs five steps. The channel assignment algorithm, used in one of the five steps, is based on three different algorithms; and (ii) the assumption that the system operates synchronously in a time-slotted mode does not hold for most of the current networks. Concerns observed in [23] include: (i) the authors assumed that all the nodes are equipped with the same number of interfaces; (ii) the log-quadratic formula is very sensitive to the interference levels; and (iii) simulation results show that the algorithm restricts those links that are neighbors of congested links from using the same channels used by congested links. At a later time, this behavior causes congestion on links that were not previously congested. In [24], the mechanism to handle channel dependencies may be very restrictive. The proposed algorithm simply aborts a reassignment if it involves any ripple effect, which may lead to discarding of good solutions.

### 4.2.3. Static assignment

Static assignment strategies [25–32] assign a channel to each interface for permanent use. They usually do not consider traffic loads, since traffic patterns may substantially vary over time. A simple channel assignment strategy called *common channel set* (CCS) is used in [28,29]. It works as follows: on every node, channel $C_1$ is assigned to interface 1, channel $C_2$ to interface 2, and so on. In [28], Draves et al. proposed an ad-hoc routing framework called mesh connectivity layer (MCL). Although MCL is flexible enough to support different channel assignment strategies, the authors applied CCS. In [29], Adya et al. presented a protocol called multi-radio unification protocol (MUP), which co-ordinates the operation of the interfaces that are tuned at startup time. The proposed architecture also uses CCS. In [25], Marina et al. formulated the channel assignment problem as a topology control problem. They developed a greedy algorithm that minimizes the maximum link conflict weight and simultaneously preserves the connectivity of the connectivity graph. Simulation results showed that the minimization of the maximum link conflict weight leads to significant gains in number of concurrent transmissions. In [30], Das et al. proposed two algorithms that also use the link conflict graph to model interference. The first algorithm minimizes the average link conflict weight, while the second minimizes the maximum link conflict weight. Both algorithms are based on an approximation algorithm for the *MAX k-CUT* problem. In [26], Tang et al. proposed a heuristic that statically binds an interface to a channel by minimizing the channel interference among links. The interference model is similar to that used in [25]. The authors also proposed an interference-aware QoS routing scheme in which the bandwidth requirement of each flow is considered. In [27], Das et al. proposed two integer linear-programming models. The objective is to maximize the number of simultaneous transmissions in the network, subject to connectivity restrictions. In [31], Kodialam et al. presented a multi-commodity network flow model used to find an upper bound of the achievable throughput for a given set of flows. The *ideal* model assumes that (i) the system operates in a synchronous time-slotted manner; (ii) the switching delay is negligible; (iii) the data rate of each link is fixed; (iv) the interference is captured based on the protocol model of interference; and (v) the flows can be routed through multiple paths. Based on this model, a greedy algorithm for static channel assignment, followed by a greedy coloring for time slot assignments was proposed. In addition, a dynamic scheme for channel assignment and scheduling was presented. In [37], the same authors proposed a simplified algorithm that was used to verify, by simulations, the theoretical results of [3]. In [32], Ko et al. presented a distributed scheme that assigns channels based on local information only. A greedy algorithm selects channels according to a channel interference cost function, which quantifies interference among channels. The algorithm leads to a stable network topology within a finite number of channel changes. The architecture requires one interface of each node be tuned to a common default channel.

*The pros of the static approaches*: they are usually simpler than other techniques, since dynamic conditions, such as variations due to traffic patterns or external interference, are not considered. A static channel assignment results in a stable network topology, which avoids the network partition and non-convergent behavior problems. Routing stability is also a benefit of having a fixed topology. A particular advantage observed in [26] results from the *N*-connected network topology (i.e., *informally*, there are at least *N* independent paths between any source–destination pair) that is guaranteed by the channel assignment algorithm. It not only preserves connectivity but also enhances the network survivability. In addition, the routing module may exploit this fact by implementing multi-path routing and load-balancing schemes. This feature is further analyzed in the context of a single-radio network by So et al. [54].

*The cons of the static approaches*: they are unable to adapt to dynamic network conditions, and therefore, should be combined with semi-dynamic approaches to achieve better performance. The network topology resulting from the static assignment can be used as a base channel assignment, while semi-dynamic reassignments are applied over time as needed. Although the preservation of the connectivity of the connectivity graph is highly desirable [25,26], poor channel assignments may result from giving excessive importance to the connectivity concern [25]. The performance of the network in terms of throughput and number of simultaneous transmissions can be improved by relaxing such constraint. Another concern of static techniques is that parameters such as data rate and transmission power are considered as fixed, despite the fact that they can vary over time. In [27], the authors assume the routing metric is *channel-unaware*. This fact contrasts with the trend of routing algorithms for WMNs, which greatly improve the network throughput by routing flows through *channel-diverse* paths, where successive hops use different channels to minimize contention among links [28]. A main disadvantage of CCS [28,29] is the inappropriate use of the channel resources. For example, consider the network topology in Fig. 15a where

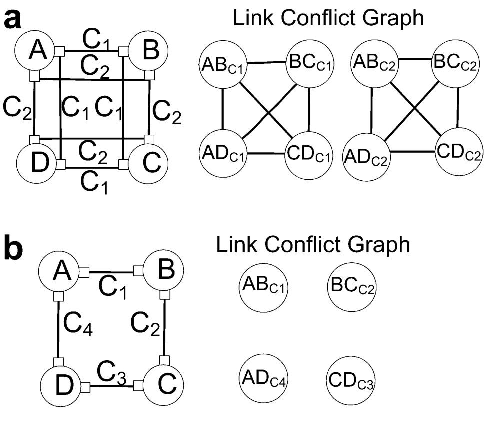

Fig. 15. Two networks with their respective link conflict graph based on the extended protocol model.

CCS is applied (on every node, the same set of channels is assigned). The corresponding link conflict graph shows that both the maximum and average link conflict weights are equal to three; i.e., each link interferes with three other links. On the other hand, Fig. 15b illustrates a solution that breaks the collision domain into four. The maximum and average link conflict weights are then both equal to zero.

### 4.2.4. Hybrid assignment

These strategies [33–36] apply a static or semi-dynamic assignment to the *fixed interfaces* and a dynamic assignment to the *switchable interfaces*. We further classified them into *receiver-fixed* [36] and *dedicated control channel* [33–35].

- Receiver-fixed

  In a receiver-fixed protocol, at least one interface is assigned a channel statically or semi-dynamically, while the other interfaces dynamically switch among the remaining channels, based on the next hop the data must be sent to. Fig. 16 illustrates the operation of a receiver-fixed protocol. The fixed interface (represented as a dark rectangle) of nodes *A*, *B*, *C* and *D* are tuned to $C_1$, $C_2$, $C_3$ and $C_4$, respectively. When node *A* has data to send to node *B*, it switches its switchable interface to channel $C_2$ and transmits the data. Given that the fixed interface of node *B* is always tuned to channel $C_2$, a *dynamic* link is established between nodes *A* and *B*. In [36], Kyasanur et al. presented a protocol called Hybrid Multi-channel Protocol (HMCP). HMCP is a distributed protocol that operates on top of the MAC layer. Periodically, each node broadcasts a *hello* packet on every channel announcing its *neighbor table* and its fixed channels. The neighbor table of a node contains the fixed channels of the neighbors of the node. The scheme allows semi-dynamic reassignments to the fixed interfaces; if a node notices that the number of nodes using the same fixed channel as itself is large, it can reassign its interface to a *less used channel*. For that purpose, each node maintains, for each channel, a counter that counts the number of nodes in its two-hop neighborhood using that channel. A particular issue with this channel reassignment mechanism is that the criterion *less used* does not account for the actual usage of the channel. In addition, the mechanism may also lead to a non-convergent behavior. Note also that, from the fact that successive hops of a path may require the switching of interfaces along the path, routing metrics must be reformulated in order to take the channel switching delay into account. For WCETT [28], a path metric based on a weighted sum, the authors suggested adding a new term to the sum to quantify the impact of the switching delay.

  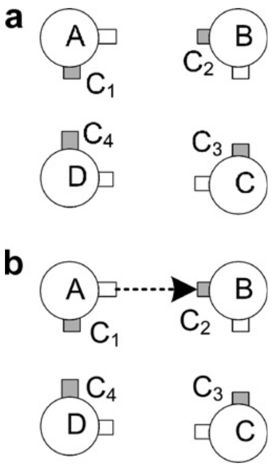

  Fig. 16. (a) Network topology where the fixed interface (dark rectangle) of *A*, *B*, *C* and *D* are tuned to $C_1$, $C_{2,}C_3$ and $C_4$, respectively. (b) Node *A* transmits data to node *B* on channel $C_2$.

  *The pros of receiver-fixed approaches*: they can be implemented with 802.11 cards. In addition, the connectivity of the connectivity graph is preserved, which avoids the network partition problem. Coming back to Fig. 16, note that, although there is no *fixed* link between any two neighbors, the routing module can still assume the existence of a fixed link between them, since they can be dynamically established. Finally, even when the switchable interfaces are dynamically handled, these protocols relieve the strict synchronization requirement.

  *The cons of receiver-fixed approaches*: the broadcast support becomes a concern, since a broadcast packet can only be received by those nodes having their respective fixed interface tuned to the channel used to send the packet. The potential solution of sending a copy of the packet on every channel may negatively impact on the performance of the protocol, since the sender has to frequently switch its interface from one channel to another. The multi-channel hidden terminal problem still remains, since when a node switches a switchable interface to a new channel, the NAV may not be up-to-date. The solution proposed in [36] (to defer for a maximum sized packet transmission time) may significantly waste bandwidth.

In addition, radios operating at different data rates (e.g., the 802.11a standard allows up to 8 different data rates) have different transmission times. Another concern is the proper interval of time a switchable interface needs to be tuned to a given channel. In Fig. 16a, if node $A$ has data to send to both, nodes $B$ and $D$, its switchable interface has to frequently switch from $C_2$ to $C_4$ and vice versa. If the interface stays on a channel, say $C_2$, for a long period, the packets addressed to node $D$ can experience excessive delays. Moreover, a large queue may be necessary to enqueue the packets waiting for being transmitted. On the contrary, if the interface stays on a channel for only a short period, the switching overhead is increased. For this example, a static or semi-dynamic channel assignment, as the one shown in Fig. 15b, may result in a better performance, since simultaneous transmissions can take place while the channel switching delay is avoided. At the same time, the complexity of managing interfaces dynamically is eliminated.

- Dedicated control channelThese protocols [33–35] are similar to the ones reviewed in Section 3.2.1. A control channel is exclusively used for control packets, while the remaining channels are used for data traffic. One interface is statically tuned to the control channel. The channel selection mechanisms presented in Section 3.1.2 can be also applied here. The remaining interfaces dynamically switch according to the channel negotiations.

  Wu et al. [33] proposed a protocol called Dynamic Channel Allocation (DCA). In DCA, channel negotiations start with the sender transmitting an RTS to the intended receiver, which replies with the corresponding CTS. DCA adds an extra control packet that is sent by the sender after receiving the CTS to announce the selected channel. In [34], DCA was extended by incorporating power control (DCA-PC). In [35], Hung et al. presented a protocol called Dynamic Private Channel (DPC). The channel negotiation mechanism of DPC is slightly different from the mechanisms previously reviewed. The initial RTS is followed by a *reply-to-RTS* (RRTS). The handshake finishes with a CTS, which may not be immediately sent after the RRTS. Simulation results show that the channel utilization increases as the number of channels increases up to four, but drops beyond four channels. This degradation is due to the blocking that occurs when the sender and receiver have already agreed on a data channel, but the latter delays sending the CTS because all its data interfaces are busy. DPC may be then improved by reducing the blocking.

  Dedicated control channel protocols for single-radio and multi-radio networks share many advantages and disadvantages. We thus refer to the Section 3.2.1 for *pros* and *cons* of these approaches. However, it is important to note that the protocols for multi-radio networks are still able to listen to the control channel while transmitting or receiving on a data channel, which avoids the deafness and multi-channel hidden terminal problems. Nevertheless, the addition of a new interface for control purposes *only* may not be justifiable, considering that even some single-radio protocols are able to obtain better performance than multi-radio dedicated control channel protocols [15]. The main problem of these protocols arises from the inefficient use of the available channels, since the control channel becomes a bottleneck under heavy loads. In addition, that channel cannot be used for data exchange, which is needed for further improvement.

## 5. Summary of channel assignment protocols and architectures

Tables 2 and 3 summarize some features of the protocols and architectures reviewed in this survey, for single-radio and multi-radio networks, respectively. The columns *Deafness problem* and *M-Channel hidden terminal problem* in Table 2 indicate whether a given protocol solves the respective problem or not. For those protocols that may suffer from such problems, the respective entry is marked as *Unsolved*. The column *Implementation* in Table 3 indicates if the proposed multi-radio scheme is implemented in a distributed or in a centralized way. For those schemes that mathematically model and solve the channel assignment problem but do not implement a real network software, the column *Implementation* suggests the more appropriate way to implement them according to our best understanding. The column *Objective* indicates the objective to be optimized. Since some schemes do not explicitly formulate an objective function, we fill out their corresponding entries with the implicit objective they try

Table 2
Comparison of multi-channel approaches for single-radio networks

| Ref | Year | Co-ordination mechanism | Channel selection mechanism | M-Channel hidden terminal problem | Deafness problem | Clock synch. | Broadcast support | MAC-changes |
|---|---|---|---|---|---|---|---|---|
| 8 | 2001 | Dedicated control channel | Measured-based. PCL power sensing | Unsolved | Unsolved | No | No | Yes |
| 9 | 2003 | Dedicated control channel | Status-based. PCL based on topological information | Unsolved | Unsolved | No | No | Yes |
| 10 | 2000 | Common hopping | Hopping sequence | Unsolved | Unsolved | Yes | No | Yes |
| 11 | 2000 | Common hopping | Hopping sequence | Unsolved | Unsolved | Yes | Yes | Yes |
| 12 | 2004 | Independent hopping | Hopping sequence | Unsolved | Unsolved | Yes | Yes | No |
| 13 | 2005 | Independent hopping | Hopping sequence | Unsolved | Unsolved | Yes | No | Yes |
| 14 | 2006 | Split phase | Status-based. PCL receiver-based | Solved | Solved | Yes | No | Yes |
| 15 | 2004 | Split phase | Status-based. PCL source–destination counter | Solved | Solved | Yes | No | Yes |
| 16 | 2003 | Split phase | Status-based. TDMA | Solved | Solved | Yes | No | Yes |
| 17 | 2007 | Split phase | TDMA | Solved | Solved | Yes | Yes | Yes |
| 14[a] | 2006 | Receiver-fixed | n/a | Solved | Unsolved | No | No | Yes |
| 18 | 1987 | Receiver-fixed | n/a | Unsolved | Unsolved | No | No | No |

[a] The proposed protocol requires an additional tone interface for signaling.

to optimize. The names of the remaining columns in both tables are self-explanatory.

Single-radio protocols are dynamic schemes, implemented in a distributed way. Although the split phase protocols [14–17] can solve the deafness and multi-channel hidden terminal problems, there are some issues that may negatively impact on their performance, namely, the (inappropriate) duration of the control and data exchange phases, the global synchronization requirement and the channel switching delay. The last two issues can also impact on hopping protocols. An advantage of hopping protocols is that they do not require channel negotiation before each data transmission. Dedicated control channel protocols do not require clock synchronization [8,9], which reduces their complexity. However, they may suffer from the control channel bottleneck as well as the deafness and multi-channel hidden terminal problems. Similarly, receiver-fixed protocols, though simple, may also suffer from the deafness and multi-channel hidden terminal problems.

Many of the single-radio protocols rely on channel selection mechanisms to find the *best* channel for the next data transmission by considering only *local* conditions at the sender and receiver, while most of the multi-radio approaches mathematically model the channel assignment problem to get a *global* solution that can be statically or semi-dynamically computed. Single-radio protocols require a finer grain control of the wireless network hardware in order to dynamically switch the interface from one channel to another between successive data transmissions. Therefore, the layer at which they are implemented may greatly impact on their performance, which suggests their implementation at layer two, as innovative MAC protocols. On the other hand, multi-radio protocols are mostly implemented at upper layers, since channel assignments are done at a different time scale (statically or semi-dynamically) and the switching of the interfaces happens much less frequently.

## 6. Open research issues

Many research issues regarding multi-channel protocols and architectures are still open. Below we list some important topics that need to be investigated.

- Multi-rate capability
  To the best of our knowledge, no multi-channel protocol that exploits or explores in-depth the

Table 3
Comparison of multi-channel approaches for multi-radio networks

| Ref | Year | Category | Subcategory | Implementation | Objective | Algorithm/model | MAC changes |
|---|---|---|---|---|---|---|---|
| 19 | 2006 | SD | External-interference-aware | Centralized | Minimization of co-channel and external interference | Breadth first search algorithm/ link conflict graph model | No |
| 20 | 2004 | SD | Load-aware | Centralized | Maximization of the capacity of heavily used links | Greedy load-aware algorithm | No |
| 21 | 2005 | SD | Load-aware | Distributed | Minimization of channel usages[a] | Heuristic distributed load-aware algorithm | No |
| 22 | 2005 | SD | Load-aware | Centralized | Maximization of the bandwidth allocation | Constant approximation algorithm/integer linear-programming | No |
| 23 | 2006 | SD | Load-aware | Distributed and Centralized | Maximization of the aggregate utility across all flow sources | Sum-Log-Quadratic method | No |
| 24 | 2006 | SD | Load-aware | Distributed | Maximization of the total throughput | Heuristic algorithm. Channel assignment based on a weighted sum metric | No |
| 25 | 2005 | S | n/a | Centralized | Minimization of the maximum link conflict weight | Greedy algorithm/link conflict graph model | No |
| 26 | 2005 | S | n/a | Centralized | Minimization of topology co-channel interference[b] | Heuristic algorithm | Yes |
| 27 | 2005 | S | n/a | Centralized | Maximization of link simultaneously activated (potential simultaneous transmissions) | Greedy algorithm – link conflict weight used as metric/integer linear-programming | No |
| 28 | 2004 | S | n/a | Centralized | n/a | Common channel set | No |
| 29 | 2004 | S | n/a | Centralized | n/a | Common channel set | No |
| 30 | 2006 | S | n/a | Centralized | 1 – Minimization of the maximum link conflict weight, 2 – Minimization of the average link conflict weight | Approximation algorithm for solving the MAX k-CUT problem/link conflict weight used as metric | No |
| 31 | 2005 | S | n/a | Centralized | Maximization of the total throughput | Greedy algorithm/multi-commodity flow formulation | Yes |
| 32 | 2007 | S | n/a | Distributed | Minimization of local interference | Greedy algorithm | No |
| 33 | 2000 | H | Dedicated control channel | Distributed | n/a | Channels for switchable interfaces are selected with a status-based, receiver- oriented selection mechanism | Yes |
| 34 | 2002 | H | Dedicated control channel | Distributed | n/a | Channels for switchable interfaces are selected with a status-based, receiver oriented selection mechanism | Yes |
| 35 | 2002 | H | Dedicated control channel | Distributed | n/a | Channels for switchable interfaces are selected with a status-based, receiver oriented selection mechanism | Yes |
| 36 | 2006 | H | Receiver-fixed | Distributed | n/a | Fixed interfaces are semi-dynamically reassigned with probability $p$ to other less used channels | No |
| 31 | 2005 | D | n/a | Centralized | Maximization of the total throughput | Greedy algorithm/multi-commodity flow formulation | Yes |

H: Hybrid; SD: Semi-dynamic; S: Static; D: Dynamic.

[a] The channel usage is proportional to the sum of the loads contributed by all the interfering neighbors that use that channel.

[b] The topology co-channel interference is equivalent to the maximum link conflict weight on the link conflict graph model.

multi-rate capability of current 802.11 wireless cards has been proposed yet. For example, the single-radio TDMA protocol presented in [17] sets the transmission rate to a fixed value of 2 Mbps. By considering only homogeneous links, the problem becomes much simpler. However, a protocol with adaptive rates can achieve better performance [44].

- Channel switching delay
  The channel switching delay is a concern for protocols that dynamically switch the interfaces between successive data transfers. How to properly set the dwell time in hopping protocols [10–13], and the control and data exchange phase durations in split phase protocols [14–17], needs to be investigated in order to optimize such protocols. In multi-radio scenarios, the channel switching delay affects hybrid protocols. An open question is whether dynamic switching in hybrid protocols is justifiable, considering the advantages that static and semi-dynamic approaches can offer, namely: (i) the complexity introduced by dynamically switching from one channel to another between successive data transfers is eliminated; (ii) the broadcast, deafness and multi-channel hidden terminal problems are avoided; (iii) the stability of the network topology favors the routing module, since current routing metrics [28] can be used without any change. On the other hand, in order to compute *good paths*, hybrid protocols require the routing module be aware of dynamic links, since multi-hop paths suffer from the delays produced by both the channel switching and the channel negotiation between successive hops. Quantitative comparisons are also desirable. The hybrid protocol proposed in [36] was only compared through simulations with the 802.11 MAC protocol working on a single channel, which clearly is not a fair comparison. It would be desirable to compare hybrid protocols with semi-dynamic protocols, or at least with multi-channel single-radio techniques (which would still be an unfair comparison).
  A study of the characterization of traffic patterns in WMNs is also needed. That study can clarify if per-packet channel switching (e.g., dynamic and hybrid techniques) is justifiable or not. In a non-interfering disjoint flow scenario, or in a lightly loaded scenario, dynamic channel switching may only introduce time overhead due to channel switching delay. For such scenarios, flow-aware switching decisions –semi-dynamic– may achieve better results. The study may be also useful for the design of techniques to achieve routing stability [45].
- Quality of service
  How to provide QoS has already been a known problem in single-channel networks. How multi-channel protocols can provide services with multiple QoS requirements while considering co-existing traffic loads is a challenging issue. As an example, voice over IP (VoIP) imposes strict bandwidth and end-to-end delay bounds in order to work properly. Multi-channel protocols need to simultaneously consider many factors such as channel switching delay, data and loss rate of individual links, intra-flow interference and inter-flow interference. How to combine them to find the best channel assignment, regarding the trade-off among different metrics, makes this problem complex. Coming back to the VoIP example, to find and choose the best trade-off solution (considering both bandwidth and end-to-end delay constraints) belonging to the feasible solution space is not trivial.
- Directional antennas
  Directional antenna technology has been identified as a promising technology in WMNs to enhance network performance [46]. Its use not only increases the spatial reuse and extends the transmission ranges but also results in better signal quality than omni-directional antennas, which in turn achieves higher data transmission rates. However, to achieve further improvement, the *directional* deafness and hidden terminal problems as well as the *multi-channel* deafness and hidden terminal problems need to be jointly solved. The directional deafness problem occurs because a node using a directional antenna is *deaf* in all the directions except for the direction in its main beam. Similarly, the directional hidden terminal problem occurs when a node is unaware of the state of a channel, when it orients its antenna to a new direction. The physical carrier sensing is also affected by directional antennas, since typical signal strength values are usually different from the values used in omni-directional antennas. This fact can in turn directly impact on the channel selection mechanisms used by multi-channel protocols. In addition, nodes need to perform virtual carrier sensing in a per-channel basis as well as per-direction basis, in order to defer only in the direction, in the channel of ongoing communication. Finally, the broadcast

problem may become more severe, since broadcast packets need to reach not only the nodes tuned to different channels but also the nodes out of the cone of coverage of the sender. Considering both multi-channel and directional antenna simultaneously is an open research issue that needs to be further investigated.

- The energy issue
  One of the most important issues in wireless networks is energy consumption. It becomes more crucial for mobile networks and sensor networks. It is not known yet if multi-channel technique can help to reduce energy consumption. On the other hand, frequently changing channels may increase energy consumption. The use of multiple interfaces also leads to higher energy requirement. Evaluating energy consumption of current proposed protocols and studying the impact of multi-channel techniques can be a major research topic.
- Capacity analysis
  The network capacity of WMNs is still a challenging topic. Although Kyasanur and Vaidya [3] extended the analysis of Gupta and Kumar [41] to characterize the impact of number of channels and interfaces on the network capacity, they still made some simplified assumptions that include: (i) completely stationary nodes; (ii) homogeneous number of interfaces per node; (iii) the optimal transmission power is known for each interface; and (iv) the global knowledge. Despite the effect of the channel switching delay is considered, an in-depth study of this topic may clarify, among others, whether multi-radio hybrid protocols have any advantage over semi-dynamic schemes or not. In addition, important techniques such as multi-input multi-output systems for which analytical models are available [53] have not been considered. Finally, new models to quantify the capacity of WMNs in presence of multiple QoS requirements are also needed.
- Integration with Vehicular Ad hoc Networks (VANETs)
  VANETs allow vehicle-to-vehicle (V2V) and vehicle-to-roadside (V2R) communications; e.g., vehicle-to-WMN. In U.S., the Federal Communications Commission allocated 75 MHz in the 5.9 GHz licensed band to VANET, supporting seven separated channels. One channel is identified as control channel, which can be used to send *safety* messages that require strict QoS in terms of delay bound. The remaining channels are called *service* channels, and can be used by *non-safety* applications [47]. Despite WMNs are suitable for V2R architectures due to their high data rate and low cost, little research has been done in this area. A challenging issue for interconnecting these two networks is the bridging function and the corresponding multi-channel protocols for both the WMN and the V2R communications. Mesh nodes with bridging capability may want to provide roadside services without compromising safety data transfer from the WMN to the vehicles or vice versa. Thus, a protocol for V2R communication needs to accomplish the following: (i) efficiently uses the seven available channels; (ii) priories the safety messages over non-safety data; (iii) provides QoS with strict delay bound (typically in the order of milliseconds [48]) for safety message transmissions; (iv) routes data from the roadside to the WMN efficiently (and potentially to other roadside spots); and (v) dynamically adapts its behavior to heterogeneous traffic patterns produced by *periodic* (e.g., buses) and *random* (e.g., taxis) vehicles. The development of new mobility models is another related topic. Marfia et al. [49] showed that the advantages of these hybrid networks can be appreciated only with accurate motion models.Two examples of current urban networks are DieselNet [50], a 40-bus network covering 150 square miles, and Shanghai Urban Vehicular Network (SUVNet) [51], a 4200-taxi network covering an area of 40 square miles. In SUVNet, more than 1000 taxis are always concentrated in the main area of Shanghai downtown (about 1 square mile). Multi-channel protocols for such networks must consider these differences in mobility and vehicular density. The mobility scenario and the MAC protocol were also reported by Campelli et al. [57] to greatly impact on the performance on V2V protocols, where the standard 802.11 MAC and ADHOC MAC [43] working on a single channel were used to evaluate and to address the issue of designing routing solutions for safety messages. DCA [33], the hybrid dedicated control channel protocol discussed in Section 4.2.4, has been also tested for V2V communications. Simulation results presented in [52] showed that, when used as a V2V protocol, DCA mainly suffers from the same problems already found in WMNs: bottleneck in the control channel and inefficient utilization of the remaining channels.

## 7. Conclusion

The use of multiple channels can substantially improve the performance of wireless mesh networks. A key issue in such networks is how to efficiently utilize the available channels. This paper has surveyed, classified and reviewed the most relevant approaches for channel assignment in WMNs. We have classified them into two different categories: for single-radio and for multi-radio networks. Single-radio protocols are dynamic protocols that try to maximize the number of simultaneous data transmissions by negotiating the most suitable channel for the next data transmission; therefore, nodes are required to dynamically switch their interface between successive data transmissions. At the same time, the protocol design needs to deal with new concerns that result from the use of multiple channels. Considering the principle of operation, we have further classified single-radio protocols into dedicated control channel, hopping, split phase and receiver-fixed. Channel assignment approaches for multi-radio networks need not implement a tight co-ordination mechanism to achieve concurrent transmissions. Simultaneous data transmissions are still achievable without a channel negotiation. This fact leads to new approaches specially designed for multi-radio WMNs. Many of them mathematically model the channel assignment problem to get a *global* optimal solution. This contrasts with single-radio protocols, which try to dynamically find the *best* channel for the next data transfer *locally* at sender and receiver locations. Based on the frequency with which channel assignments are done, we have classified multi-radio approaches into dynamic, semi-dynamic, static and hybrid. Finally, we have also pointed out important open research issues that need to be investigated.

## Acknowledgements

We thank the anonymous reviewers for their valuable comments. This research was partially supported by National Science Foundation of USA under Grant CNR-0626380, Natural Science Foundation of China under Grants Nos. 60573138 and 60773091, National Basic Research Program of China (973 Program) under Grant No. 2006C B 303000, and National 863 Program of China under Grant No. 2005aa122320.

## References

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

[2] IEEE Std. 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999. Available at <http://standards.ieee.org/getieee802/802.11.html>.

[3] P. Kyasanur, N. Vaidya, Capacity of multi-channel wireless networks: Impact of number of channels and interfaces, in: 11th ACM International Conference on Mobile Computing and Networking (MobiCom 2005), August 2005, Cologne, Germany.

[4] K. Jain, J. Padhye, V.N. Padmanabhan, L. Qiu, Impact of interference on multi-hop wireless network performance, in: 9th Annual International Conference on Mobile Computing and Networking (Mobicom 2003), September 2003, San Diego, California – USA.

[5] D. Johnson, D. Maltz, J. Broch, DSR: The dynamic source routing protocol for multihop wireless ad hoc networks, in: Ad Hoc Networking, Addison-Wesley, 2001, pp. 139–172 (Chapter 5).

[6] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On-Demand Distance Vector (AODV) Routin, IETF RFC 3561, July 2003.

[7] J. Mo, H. So, J. Walrand, Comparison of multi-channel mac protocols, in: 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Montreal, Quebec – Canada, 2005.

[8] N. Jain, S. Das, A. Nasipuri, A multichannel csma mac protocol with receiver-based channel selection for multihop wireless networks, in: 10th International Conference on Computer Communications and Networks (ICCCN 2001), Scottsdale, Arizona – USA, 2001.

[9] J. Li, Z. Haas, M. Sheng, Y. Chen, Performance evaluation of modified ieee 802.11 mac for multi-channel multi-hop ad hoc network, in: 17th International Conference on Advanced Information Networking and Applications (AINA), Xi'an, China, March 2003.

[10] A. Tzamaloukas J. Garcia-Luna-Aceves, Channel-hopping multiple access, in: IEEE International Conference on Communications (ICC'2000), New Orleans, Louisiana – USA, June 2000.

[11] A. Tzamaloukas J. Garcia-Luna-Aceves, Channel-hopping multiple access with packet trains for ad hoc networks, in: IEEE Device Multimedia Communications (MoMuC'00), Tokyo, Japan, 2000.

[12] P. Bahl, R. Chandra, J. Dunagan. SSCH: Slotted seeded channel hopping for capacity improvement in ieee 802.11 adhoc wireless networks, in: 10th ACM International Conference on Mobile Computing and Networking (MobiCom 2004), Philadelphia, Pennsylvania – USA, 2004.

[13] H. So, J. Walrand, McMAC: A multi-channel mac proposal for ad-hoc wireless networks, Technical Report, April 2005. Available at <http://www.cs.berkeley.edu/~so/pubs/mcmac_desc.pdf>.

[14] R Maheshwari, H. Gupta, S. Das, Multichannel mac protocols for wireless networks, in: 3rd IEEE Communication Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, Reston, Virginia – USA, September 2006.

[15] J. So, N.H. Vaidya, Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single

transceiver, in: 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc 2004), Tokyo – Japan, May 2004.

[16] J. Chen, S. Sheu, C. Yang, A new multichannel access protocol for ieee 802.11 ad hoc wireless lans, in: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2003), Beijing, China, 2003.

[17] J. Zhang, G. Zhou, C. Huang, S. Son, J. Stankovic, TMMAC: An energy efficient multi-channel mac protocol for ad hoc networks, in: ACM SIGMO BILE Mobile Computing and Communications Review (ACM MC2R), Special Issue on Localization Technologies and Algorithms, January 2007.

[18] N. Shacham, P. King, Architectures and performance of multichannel multihop packet radio networks, IEEE Journal on Selected Areas of Communication SAC-5 (6) (1987) 1013–1025.

[19] K. Ramachandran, E. Belding, K. Almeroth, M.Buddhikot, Interference-aware channel assignment in multi-radio wireless mesh networks, in: 25th Conference on Computer Communications (Infocom 2006), Barcelona – Spain, April 2006.

[20] A. Raniwala, K. Gopalan, T. Chiueh, Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks, Mobile Computing and Communications Review 8 (2) (2004) 50–65.

[21] A. Raniwala, T. Chiueh, Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network, in: 24th Conference on Computer Communications (Infocom 2005), Miami, Florida – USA, March 2005.

[22] M. Alicherry, R. Bhatia, L. Li, Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks, in: 11th ACM International Conference on Mobile Computing and Networking (MobiCom 2005), Cologne, Germany, 2005.

[23] A. Mohsenian, V. Wong, Joint optimal channel assignment and congestion control for multi-channel wireless mesh networks, in: IEEE International Conference on Communications 2006 (ICC 2006), Istanbul, Turkey, June 2006.

[24] H. Wu, F. Yang, K. Tan, J. Chen, Q. Zhang, Z. Zhang, Distributed channel assignment and routing in multi-radio multi-channel multi-hop wireless networks, Journal on Selected Areas in Communications 24 (November) (2006) 1972–1983.

[25] M. Marina, S. Das, A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks, in: 2nd International Conference on Broadband Networks (Broadnets 2005), Boston, Massachusetts – USA, October 2005.

[26] J. Tang, G. Xue, W. Zhang, Interference-aware topology control and QoS routing in multi-channel wireless mesh networks, in: 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc 2005), Urbana, Champaigne, Illinois – USA, 2005.

[27] A. Das, H. Alazemi, R. Vijayakumar, S. Roy, Optimization models for fixed channel assignment in wireless mesh networks with multiple radios, in: 2nd IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, California – USA, September 2005.

[28] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: 10th ACM International Conference on Mobile Computing and Networking (Mobicom 2004), Philadelphia, Pennsylvania – USA, September 2004.

[29] A. Adya, P.B ahl, J. Padhye, A. Wolman, L. Zhou, A multiradio unification protocol for ieee 802.11 wireless networks, in: International Conference on Broadband Networks (BroadNets), San Jose, California – USA, October 2004.

[30] A. Das, R. Vijayakumar, S. Roy, Static channel assignment in multi-radio multi-Channel 802.11 wireless mesh networks: issues, metrics and algorithms, in: 49th IEEE Global Telecommunications Conference (Globecom), San Francisco, California – USA, November 2006.

[31] M. Kodialam, T. Nandagopal, Characterizing the capacity region in multi-radio multi-channel wireless mesh networks, in: 11th ACM International Conference on Mobile Computing and Networking (MobiCom 2005), Cologne, Germany, 2005.

[32] B. Ko, V. Misra, J. Padhye, D. Rubenstein, Distributed channel assignment in multi-radio 802.11 mesh networks, in: IEEE Wireless Communications and Networking Conference (WCNC 2007), 2007, Hong Kong.

[33] S. Wu, C. Lin, Y. Tseng, J. Sheu. A new multi-channel mac protocol with on-demand channel assignment for multi-hop mobile ad hoc networks, in: International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '00), Dallas, Texas – USA, December 2000.

[34] S. Wu, C. Lin, Y. Tseng, C. Lin, J. Sheu, A multi-channel mac protocol with power control for multi-hop mobile ad-hoc networks, The Computer Journal 45 (1) (2002) 101–110.

[35] W. Hung, K. Law, A. Leon-Garcia, A dynamic multichannel mac for ad-hoc lan, in: 21st Biennial Symposium on Communications, Kingston, Canada, June 2002, pp. 31–35.

[36] P. Kyasanur, N. Vaidya, Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks, SIGMOBILE Mobile Computing and Communications Review 10 (1) (2006) 31–43.

[37] M. Kodialam, T. Nandagopal, On the capacity region of multi-radio multi-channel wireless mesh networks, in: IEEE Workshop on Wireless Mesh Networks (WiMesh 2005), Santa Clara, California – USA, September 2005.

[38] Nordic Semiconductor, <http://www.nordicsemi.no/index.cfm?obj=product&act=display&pro=89>.

[39] Maxim 2.4 GHz 802.11b Zero-IF Transceivers, <http://pdfserv.maxim-ic.com/en/ds/MAX2820-MAX2821.pdf>.

[40] J. Elson, L. Girod, D. Estrin, Fine-grained network time synchronization using reference broadcast, in: 5th Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, Massachusetts – USA, December 2002.

[41] P. Gupta, P. Kumar, The capacity of wireless networks, IEEE Transactions on Information Theory 46 (2) (2000).

[42] X. Yang, N. Vaidya, Priority scheduling in wireless ad hoc networks, in: 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002), Laussane, Switzerland, 2002.

[43] F. Borgonovo, A. Capone, M. Cesana, L. Fratta, Adhoc mac: a new mac architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services, Wireless Networks (WINET) 10 (4) (2004) 359–366.

[44] H. Zhai, Y. Fang, Impact of routing metrics on path capacity in multi-rate and multi-hop wireless ad hoc networks, in: 14th IEEE International Conference on Net-

work Protocols (ICNP'06), Santa Barbara, California – USA, November 2006.

[45] K. Ramachandran, I. Sheriff, E. Belding, K. Almeroth, Routing stability in static wireless mesh networks, in: Passive and Active Measurement Conference, Belgium, April 2007.

[46] S. Das, H. Pucha, D. Koutsonikolas, Y. Hu, D. Peroulis, DMesh: incorporating practical directional antennas in multichannel wireless mesh networks, IEEE Journal on Selected Areas in Communications (JSAC) 24 (11) (2006), Special issue on Multi-Hop Wireless Mesh Networks.

[47] FCC Report and Order: FCC-03-324, February 2004.

[48] T. Mak, P. Laberteaux, R. Sengupta, A multi-channel vanet providing concurrent safety and commercial services, in: 2nd ACM International Workshop on Vehicular Ad Hoc Networks, Germany, 2005.

[49] G. Marfia, G. Pau, E. Giordano, E. de Sena, M. Gerla, Vanet: On mobility scenarios and urban infrastructure. A case study, in: Move'07, Anchorage, Alaska – USA, May 2007.

[50] http://prisms.cs.umass.edu/dome/.

[51] H. Huang, P. Luo, M. Li, D. Li, X. Li, W. Shu, M. Wu, Performance evaluation of suvnet with real-time traffic data, IEEE Transactions on Vehicular Technology 56 (6) (2007).

[52] X. Zhang, H. Su, H. Chen, Cluster-based multi-channel communications protocols in vehicle ad-hoc networks, IEEE Wireless Communications Magazine 13 (5) (2007) 44–51.

[53] P. Almers, E. Bonek, A. Burr, N. Czink, M. Debbah, V. Degli-Esposti, H. Hofstetter, P. Kyösti, D. Laurenson, G. Matz, A. Molisch, C. Oestges, H. Özcelik, Survey of channel and radio propagation models for wireless mimo systems, EURASIP Journal on Wireless Communications and Networking 2007 (2007).

[54] J. So, N. Vaidya, Load-balancing routing in multichannel hybrid wireless networks with single network interface, IEEE Transactions on Vehicular Technology 55 (3) (2006) 806–812.

[55] U. Lee, S. Midkiff, Channel distribution fairness in multi-channel wireless ad-hoc networks using a channel distribution index, in: 25th IEEE International Performance, Computing, and Communications Conference (IPCCC 2006), Phoenix, Arizona – USA, April 2006.

[56] S. Damodaran, K. Sivalingam, Scheduling in wireless networks with multiple transmission channels, in: 7th International Conference on Network Protocols (ICNP 1999), Toronto – Canada, November 1999.

[57] L. Campelli, M. Cesana, R. Fracchia, Directional broadcast forwarding of alarm messages in VANETs, in: Proc. of IEEE/IFIP WONS 2007, Obergurgl, Austria, January 2007.

[58] Meshdynamics, <http://www.meshdynamics.com/third_generation.html>.

**Jorge Crichigno** received his B.S. in electrical engineering from the Catholic University of Asuncion, Asuncion, Paraguay, in 2004. He is currently a Ph.D. student in the Department of Electrical and Computer Engineering, the University of New Mexico. From March to July of 2007 he was visiting the School of Electronic, Information and Electrical Engineering at Shanghai Jiao Tong University. His current research interests include mobile and wireless mesh networks.

**Min-You Wu** is an IBM Chair Professor in the Department of Computer Science and Engineering at Shanghai JiaoTong University. He serves as the Chief Scientist at Grid Center of Shanghai JiaoTong University. His research interests include grid computing, wireless networks, sensor networks, overlay networks, multimedia networking, parallel and distributed systems, and compilers for parallel computers. He has published over 110 journal and conference papers in the above areas. He is a member of ACM and a senior member of IEEE.

**Wei Wennie Shu** received the Ph.D. degree from the University of Illinois at Urbana, Champaign. Since then, she worked at Yale University, the State University of New York at Buffalo, and University of Central Florida. She is currently an Associate Professor in the Department of Electrical and Computer Engineering, the University of New Mexico. Her current interests include dynamic scheduling, resource management, multimedia networking, distributed systems, and sensor networks. She is a member of ACM and a senior member of IEEE.