Computer Networks 53 (2009) 2225–2240

Contents lists available at ScienceDirect

Computer Networks

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

Role and channel assignments for wireless mesh networks using hybrid approach ☆

Andy An-Kai Jeng, Rong-Hong Jan *

Department of Computer Science, National Chiao Tung University, Hsinchu 30050, Taiwan, ROC

ARTICLE INFO

Article history:
Received 7 September 2008
Received in revised form 20 January 2009
Accepted 29 March 2009
Available online 17 April 2009

Responsible Editor: L. Lenzini

Keywords:
Wireless mesh network
Multi-channel multi-interface
Channel assignment
Role assignment
NP-hardness
Greedy algorithm

ABSTRACT

The wireless mesh network has been considered one of the most promising techniques to extend the broadband access to the last mile. To utilize multiple channels on more than one interface, a number of approaches have been proposed. In particular, the hybrid strategy that combines the benefits of high channel diversity and low coordination cost has seen growing interest in recent studies. In this paper, we define two optimization problems, named the role assignment and the semi-fixed channel assignment, to characterize the unique feature in the hybrid strategy. We proved that the two problems are $\mathcal{NP}$-hard even if the transmission ranges of interfaces are equal. In order to solve our problems in reasonable time, we design efficient algorithms. For the role assignment problem, we give an 1/2-approximate algorithm to find a nearly optimal solution. For the semi-fixed channel assignment problem, a heuristic algorithm, based on transferring from a coloring-based problem, is proposed. Experimental results show that optimizing the defined problems is indeed beneficial to improve the network throughput, and the proposed algorithms are significantly superior to existing methods.

1. Introduction

The demand on wireless broadband access has been continually burgeoning in the recent years. Particularly, the wireless mesh network (WMN) is considered one of the most promising techniques for extending broadband access to the last mile [1]. The WMN consists of a set of mesh access points (MAPs). A mobile station can access to the network by connecting to a nearby MAP. Each MAP acts as a wireless router to forward traffic hop-by-hop to destinations. Thus, by deploying in this fashion, a backhaul network can be easily built up without wired connection.

Unlike the mobile ad hoc network (MAENT), nodes in WMNs are usually static with continuous power supplies. Hence, the issues about mobility and energy efficiency are less critical. Instead, the capacity of the backhaul is a major concern. The backhaul has to provide sufficient bandwidth to support traffic between MAPs and Internet gateways as well as communications between MAPs themselves. To increase the capacity, one approach is to utilize multiple channels [2,3]. The IEEE 802.11 b/g and 802.11a standards provide up to 3 and 12 non-overlapped channels, respectively, in 2.4 GHz and 5 GHz spectrums. Nodes within the transmission range of each other can turn their interfaces to different non-overlapped channels to avoid interference. Besides, the throughput can be further enhanced by equipping nodes with multiple interfaces [4,5]. That is, a node with two or more interfaces can perform simultaneous transmissions and/or receptions on different channels to increase throughput in parallel.

Ideally, the capacity can be multiplied by H times if there are H interfaces and channels available to each node. However, in practice, it is too expensive to equip nodes with as many interfaces as the number of channels (recall that IEEE 802.11a has 12 channels). Therefore, how to

☆ This research was supported in part by the National Science Council, Taiwan, ROC, under Grants NSC97-2752-E-009 -005-PAE and NSC 97-2219-009-006.

* Corresponding author. Tel.: +886 3 5731637; fax: +886 3 5721490.
E-mail address: rhjan@cis.nctu.edu.tw (R.-H. Jan).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.03.020

exploit multiples channels with a few interfaces (usually 1–4) by each node is a key issue. On the other hand, although using diverse channels is attractive, whenever two neighboring nodes want to communicate, they have to ensure that some of their interfaces are on a common channel; otherwise, they cannot detect signals from each other.

To address these issues, a number of studies have been conducted in the literature [6]. According to the classification in Skalli et al. [7] and Kyasanur and Vaidya [8], existing approaches can be categorized into the *static*, *dynamic*, and *hybrid* strategies.

In the static strategy [5,9–11], each interface is fixed on a channel permanently or for a long period of time. If a node *A* wants to communicate with a neighboring node *B*, they must have some interfaces fixed on the same channel. Then, *A* can send packets (e.g. RTS/CTS/DATA) directly to *B*, without an additional control process to find a common channel. However, it also limits the ability of using diverse channels. For example, as shown in Fig. 1a, there are four channels ch_1, ch_2, ch_3 and ch_4, and *A* is relaying packets from *S* to *B* on ch_2. If ch_2 is now interfered by other transmissions, *A* cannot utilize other channel except ch_1 and ch_2. In other words, the number of channels that can be utilized by a node is limited by the number of its interfaces. In addition, if there is no common channel shared by two adjacent nodes (e.g., *A* and *C*), their traffic has to be relayed through a longer path (e.g., A, B, T, C). Even worse, the network will be partitioned if there is no alternative path.

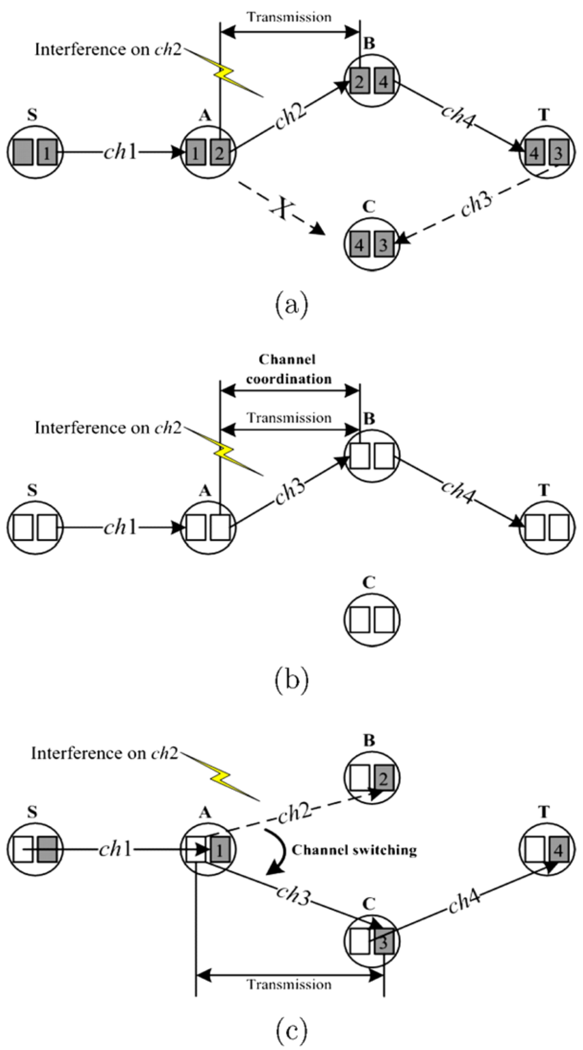

Fig. 1. (a) Static strategy; (b) dynamic strategy and (c) hybrid strategy.

In contrast, the dynamic strategy allows each interface to switch its channel from time to time to exploit the maximum channels diversity [12–15]. For example, in Fig. 1b, node *A* can turn its interface to a less interfered channel (e.g., ch_3), if it observes that the current signal-to-noise ratio (SINR) is below a threshold. Nonetheless, since those assigned channels are not fixed, before a transmission, a coordination mechanism is required to find a common channel for the sender–receiver pair. In [13], all nodes have to periodically rendezvous on a default channel to negotiate data channels for the hereafter transmissions. Another way is to use pseudo-random channel hopping sequences [14]. These mechanisms may spend considerable bandwidth for exchanging control packets or depend on sophisticated time synchronization. Therefore, the implementation is harder.

The hybrid strategy [8,16–19] combines the benefits from both the static and dynamic strategies. Herein, each interface is either *fixed* or *switchable*. Like the static strategy, a fixed interface will stay on a channel permanently or for a long period of time. On the other hand, a switchable interface can switch among different channels. For a transmission, the sender turns one of its switchable interface to a channel that is fixed on some interface of the receiver, and then starts transmitting on that channel. In this way, a node can utilize diverse channels via its switchable interface(s). For example, assume that node *S* wants to send a message to node *T* in Fig. 1c. First, node *S* sends the message to node *A*. If *A* observes that ch_2 is interfered, it can just switch its switchable interface to ch_3 which is the channel of the relaying node *C*'s fixed interface. More importantly, since channels of fixed interfaces are rarely changed, switching can be made immediately without any coordination.

Overall, the hybrid strategy is attractive due to its practicality (the implementation of channel switching is easier comparing with channel coordination) and the flexibility of using diverse channels. However, so far, only a few studies have been made for this strategy. There was no research thoroughly exploring the unique feature about coexisting the dynamic channel switch and fixed channel usage. For this reason, we attempt to further investigated into this strategy. There are three major contributions in this paper. First, we define two optimization problems for the hybrid strategy as follows.

(1) *Role assignment problem*: Recall that an interface can be either fixed or switchable. Given a set of interfaces, the *role assignment problem* is to decide which one should be fixed and which one should be switchable. The problem was not discussed before, because the previous works usually assumed that

each node has only two interfaces with the same coverage. In this setting, there is no choice but assigning one to be fixed and another to be switchable. But when there are more than two interfaces or having varied transmission ranges, a various assignment would result in a completely different topology. In order to preserve the original topology, we aim to maximize the total number of switching pairs among interfaces, which can help nodes to find more paths to avoid interference, balance traffic load, and increase throughput. In addition, the connectivity between nodes should be guaranteed.

(2) *Semi-fixed channel assignment problem*: The channel assignment problem has been extensively studied for the static and dynamic strategies [7]. But, the concern for the hybrid strategy is quite different. To explain it, let us see an example in Fig. 1c. No matter which neighbor of A (i.e. B, C, or S) wants to transmit to A, it has to switch its switchable interface to ch_1 that is fixed on A. Now assume that some nodes surrounding to C (not drawn here) are also contending for ch_1. To avoid interference, intuitively, A should change its fixed interface to other channel before receiving from C. However, it is not appropriate to change the channel of a fixed interface frequently; otherwise, the node may need to spend more time and bandwidth to update its fixed channel to nearby nodes. Moreover, if the updated channel was not correctly received by all nearby nodes, a miss matching channel switching may occur. As shown in Fig. 1c, assume that A has changed its fixed interface from ch_1 to ch_2 and broadcasted ch_2 to its neighbors. If S did not detect this event due to interference, S may still switch to ch_1 and send RTS on this channel. As a result, the RTS will not be received by A, which in turn incurs additional cost for retransmissions. For these reasons, the channel assigned on each fixed interface should be rarely changed and meanwhile should satisfy the possible channel switching from nearby nodes. We called such problem as the *semi-fixed channel assignment*. To characterize this problem, we define a cost metric that measures the average number of links interfered by any possible channel switching. The cost metric is not specific to any traffic pattern, which can help nodes to adapt to time-varying traffic demands without changing to the assigned result. This is the most desired feature in the hybrid strategy. The details of the two problems will be explained in Section 3.2.

Second, we analyze the complexity issue. It can be shown that the two defined problems are $\mathcal{NP}$-hard even if the transmission ranges are equal. Since the problems are very difficult, the third contribution of this paper is to design efficient algorithms: For the role assignment problem, we give an 1/2-approximate algorithm to find nearly optimal solution. On the other hand, a heuristic algorithm, based on transferring from a coloring-based problem, is presented for solving the semi-fixed channel assignment problem. Experimental results show that optimizing the two problems is indeed beneficial to improve the performance and our algorithms are significantly superior to existing methods.

The rest of this paper is organized as follows. In Section 2, we review articles related to our study. Next, the network model and two concerned problems are formally defined in Section 3. In Sections 4 and 5, we present the complexity and algorithm, respectively, for the two defined problems. Section 6 conducts simulations to evaluate our designs. Conclusion remarks and future research are given in the last section. The proofs of theoretic properties are detailed in Appendix A.

2. Related works

The hybrid strategy first appears in [8]. It is similar to an approach in [3], where each node has one interface fixed on a common channel for exchanging control packets, and the other interfaces are switched among the remaining channels for data transmissions. The discrepancy is that the hybrid strategy allows fixed interfaces operating on different channels instead of a single one, which can avoid performance saturation from the common channel.

A hybrid multi-channel protocol (HMCP) was proposed in [16,17]. This protocol consists of two parts: The first part handles MAC issues, including queuing, switching, and broadcasting, in the hybrid strategy. Two timers are set to avoid the cost from frequent channel switching and hidden terminal nodes. The second part is a distributed channel assignment algorithm. Each node maintains a table to record the channels being used by its neighbors. Based on this table, a node periodically checks the number of other nodes using the same channel as itself. If the number is larger than average, the node will adjust its fixed interface to a less used channel with a probability p, and advertise this information through periodic "Hello" message. This approach can distribute channels equally on neighboring fixed interfaces. An abstraction module was implemented in [18]. It provides the requisite kernel support for this protocol. In addition, a multi-channel routing protocol (MCR) for the hybrid strategy was proposed in [16]. This protocol incorporates switching costs of interfaces and expected transmission time of links to select routing path. It can reduce the latency from switch and link loss.

Nonetheless, the HMCP protocol has no guarantee on its convergency; the channels of fixed interfaces may constantly change if the numbers among nodes recursively depend on each other. Besides, since nodes are not on the same channel, a Hello message has to be broadcasted on every channel to ensure that all neighbors will receive it, which will incur considerable overhead and latency. The relationship between the required number of interfaces and the available channels was also discussed in [8]. But, they do not explicitly point out how to assign the role for each interface.

Consider other major channel assignment protocols under the static and dynamic strategies [5,11,24–26]. A flow-based approach, named the load-aware channel assignment (LACA) was proposed in [5]. This protocol can iteratively adjust channels and routing pathes among nodes to allocate sufficient bandwidth to each wireless link for a given traffic profile. Similarly, a protocol, named the balanced-static channel assignment (BSCA), was designed to maximize the bandwidth allocated to each traffic aggregation point subject to fairness constraint [11]. Most recently, a flow-based approach in conjunction with rate control (FCRA) was presented in [24]. The FCRA jointly assigns links with proper channels and transmission rates to make a given set of flow rates on links schedulable [24]. Although these flow-based approaches can optimize the performance for a given traffic profile, they may not be suitable for dynamic environments, where the traffic demand cannot be known a prior. The common channel assignment (CCA) [25] and connected-low-interference channel assignment (CLICA) [26] are two flow-independent approaches. The CCA statically assigns nodes with a common set of channels. By doing so, the connectivity between nodes can be easily preserved. However, it also limits the number of channels to be used by each node. For example, with 2 interfaces, only 2 channels can be utilized by all nodes. The CLICA improves the CCA by allowing each node to operate on a different set of channels and at the same time guarantees the node connectivity. But, since the CLICA is also a static approach, the number of channels that can be used by a node is still limited by the number of its interfaces. By contrast, the hybrid strategy allows each node to switch to diverse channels. To the best of our knowledge, there was no research exploring the concerned problems in this paper.

3. Network model and problems

In this section, we formally define the network model and the two concerned problems under study.

3.1. Network model

The network consists of a set $V_{\mathrm{N}} = \{u_1, u_2, \ldots, u_n\}$ of n static mesh nodes. Each node u_i has R_i interfaces, where $R_i \geqslant 2$. Besides, H orthogonal channels are available to each interface. Let u_{ir} stand for the rth interface of a node u_i. The transmission power (and the corresponding transmission/interference ranges) of each u_{ir} is fixed, i.e. no power control. As shown in Fig. 2a, the underlying topology under the same channel can be modeled as a digraph $G_{\mathrm{T}} = (V_{\mathrm{I}}, E_{\mathrm{T}} \cup E'_{\mathrm{T}})$, named the *transmission graph*, in which $V_{\mathrm{I}} = \{u_{ir} | i = 1, 2, \ldots, n, r = 1, 2, \ldots, R_i\}$, representing the set of interfaces, a directed edge $u_{ir}u_{jt} \in E_{\mathrm{T}}$ if and only if u_{ir} can transmit data to u_{jt}, and a directed edge $u_{ir}u_{jt} \in E'_{\mathrm{T}}$ if and only if u_{ir} can simply interfere with u_{jt} but cannot transmit data to u_{jt}.

We assume that the IEEE 802.11 DCF (or other contention-based MAC) is employed to avoid hidden terminal nodes. To support it, the link between any sender–receiver pair of interfaces should be bidirectional; otherwise, the RTS–CTS–DATA–ACK sequence cannot be successfully exchanged. Moreover, there is no need to transmit on the air between any two interfaces of the same node, since in this case the traffic can be bridged inside by hardware circuits. With these considerations, given a G_{T}, all communicable links can be represented as a subgraph $G_{\mathrm{C}} = (V_{\mathrm{I}}, E_{\mathrm{C}})$ of G_{T}, named the *communication graph*, see Fig. 2b, where an edge $u_{ir}u_{jt} \in E_{\mathrm{C}}$ if and only if $u_{ir}u_{jt} \in E_{\mathrm{T}}, u_{jt}u_{ir} \in E_{\mathrm{T}}$, and $i \neq j$.

Throughout this paper, we denote $N_{ir}(G)$ and $E_{ir}(G)$, respectively, as the sets of adjacent vertices and incident edges of some u_{ir} in a graph G. In addition, the sets of outgoing and incoming edges (vertices) are distinguished by the superscripts of "−" and "+", correspondingly, i.e. $N_{ir}(G) = N^{+}_{ir}(G) + N^{-}_{ir}(G)$, and $E_{ir}(G) = E^{+}_{ir}(G) + E^{-}_{ir}(G)$. The terms "link" and "edge" will be used interchangeably. Besides, to differentiate from the graphic term "vertex", the term "node" is referred to "mesh node".

3.2. Role assignment problem

Given a set V_{I} of interfaces, a *role assignment* $\rho = (V_{\mathrm{F}}, V_{\mathrm{S}})$ is a disjointed partition of V_{I}, where V_{F} and V_{S} correspond to the sets of fixed and switchable interfaces, respectively. For any two interfaces u_{ir} and u_{jt}, u_{ir} can switch to the channel of u_{jt} only if u_{ir} is switchable and u_{jt} is fixed. So, given a role assignment ρ, all possible switching pairs among interfaces in a communication graph G_{C} can be represented as a digraph $G_{\mathrm{S}} = (V_{\mathrm{I}}, E_{\mathrm{S}})$, where a directed edge $u_{ir}u_{jt} \in E_{\mathrm{S}}$ if and only if $u_{ir}u_{jt} \in E_{\mathrm{C}}, u_{ir} \in V_{\mathrm{S}}$, and $u_{jt} \in V_{\mathrm{F}}$. We named G_{S} the *switchability graph on interfaces* and any edge in E_{S} the *switchable link*.

As shown in Fig. 2c, after assigning the roles of interfaces, some communicable links in G_{C} are no longer in G_{S}. In order to preserve the original topology of G_{C}, our goal is to maximize the *total number of switchable links* (*TSL*) in the resulting G_{S}. There are three reasons:

1. In comparison with the static strategy, the hybrid strategy allows each node to transmit to different targets using diverse channels to avoid interference. For example, node u_2 can transmit to u_3 using the channel fixed on either u_{31} or u_{32}, depending on real-time conditions. Therefore, if channels are properly arranged for fixed interfaces, a larger *TSL* can increase the chance of switching to non-interfered channels.
2. Increasing the *TSL* can also help nodes to forward packets on alternative routes to balance traffic load. For instance, with the link of $u_{51}u_{62}$, node u_5 can communicate with u_1 by relaying through u_6 whenever u_3 is overloaded.
3. For two adjacent nodes, if there are multiple switchable links between them, such as links $u_{51}u_{31}$ and $u_{52}u_{32}$ between u_5 and u_3, their throughput can be multiplied by transmitting on different channels in parallel.

On the other hand, the network connectivity should be guaranteed. To be precise, consider an role assignment ρ, we define $\overline{G}_{\mathrm{S}} = (V_{\mathrm{N}}, \overline{E}_{\mathrm{S}})$ as the *switchability graph on mesh nodes*, where a directed edge $u_iu_j \in \overline{E}_{\mathrm{S}}$ if and only if there is a switchable interface u_{ir} on u_i and a fixed interface u_{jt} on u_j such that $u_{ir}u_{jt} \in E_{\mathrm{S}}$. In other words, an edge $u_iu_j \in \overline{E}_{\mathrm{S}}$ means that node u_i can initiate a transmission with node u_j, see Fig. 2d. We say that a role assignment ρ is *feasible* if and only if the corresponding $\overline{G}_{\mathrm{S}}$ is strongly

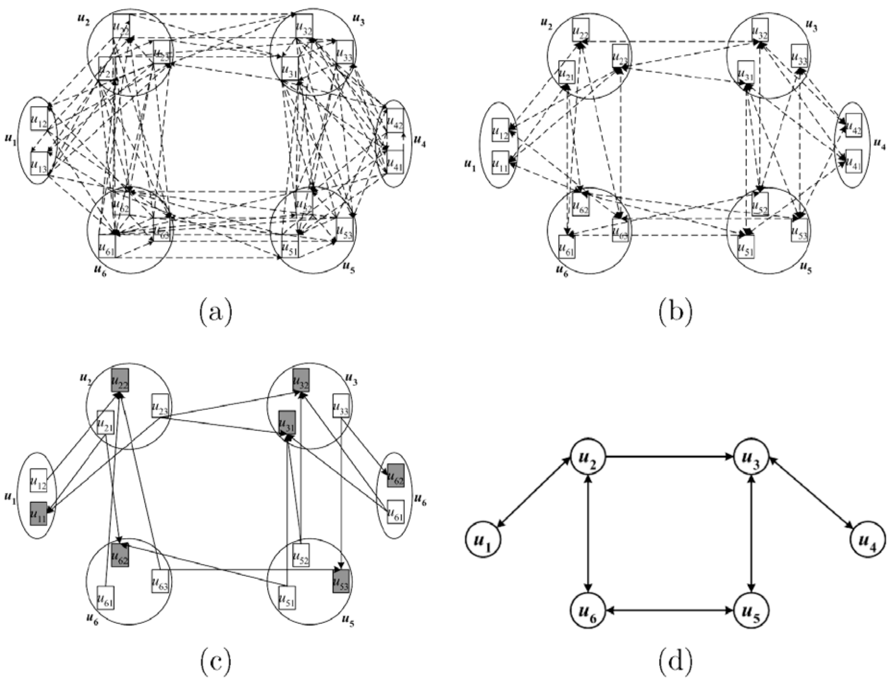

Fig. 2. (a) Transmission graph; (b) communication graph; (c) switchability graph on interfaces and (d) switchability graph on mesh nodes.

connected, i.e. for any two mesh nodes u_i and u_j in V_N, there is a path from u_i to u_j, and vice versa. Accordingly, the role assignment problem can be now defined as follows.

Definition 1 (*Max TSL*). Given a communication graph $G_C = (V_I, E_C)$, find a feasible role assignment $\rho = (V_F, V_S)$ of V_I such that $|E_S|$ is maximized.

Note that in the original design of the hybrid strategy [8], if two fixed interfaces are on the same channel, their transmission is also allowed. However, our preliminary experiments show that the performance has little change when prohibiting nodes from transmitting in such case. There are two reasons: First, as channels are uniformly spread on fixed interfaces, the chance of finding such a communication pair is lower. Second, most of fixed interfaces already spend a large portion of time to receive from other switchable interfaces so that they nearly have no time to do transmitting, especially when traffic load is heavy. On the other hand, if this case is considered, the channels assigned on fixed interfaces will change the topology which however is further an influential factor of the assignment of channels. Therefore, the channel assignment problem, as defined later, would become very complicated and intractable. For these reasons, we do not consider this circumstance in our study.

3.3. Semi-fixed channel assignment problem

As a switchable link is active for some transmission, it may conflict to other links so that they cannot be active simultaneously. Furthermore, when there are many links conflicting to each other, the throughput will degrade significantly. Therefore, our primary goal in this channel assignment problem is to minimize the possible conflict between switchable links. Now let us consider two switchable links e and e'. As e is active, there are two circumstances that e' cannot be active at the same time.

(1) *Interface blocking*: This happens when the required interface of e' is occupied by e. See for example in Fig. 3a. As u_8u_1 is active, $u_9u_1, u_{12}u_1$, and $u_{13}u_1$ should be silent, since they share a common fixed interface u_1. On the other hand, u_8u_3 cannot be active; otherwise, the channel of u_8 has to be switched from ch_1 to ch_2. Notice that although u_4 is on the same channel of u_8u_1, a collision would occur when u_8 receives ACKs from both u_4 and u_1 (recall that the 802.11DCF is used). Hence, u_8u_4 cannot be active. In these cases, we say e' is *blocked* by e, or say, e' is a *blocked link* of e. Let $e = u_{ir}u_{jt}$. All blocked links of e can be defined by

$$BL(e) = E^-_{ir}(G_S) \cup E^+_{jt}(G_S) - \{e\}.$$

An important feature of this circumstance is that any blocked link is determined as long as the assignment ρ is given, no matter what channels are assigned on V_F.

(2) *Co-channel interference*: As e is active, the RTS–CTS–DATA–ACK sequence has to be exchanged between u_{ir} and u_{jt}. Therefore, all other interfaces within the interference range of u_{ir} and/or u_{jt}, i.e.

$$PIN(e) = N^-_{ir}(G_T) \cup N^-_{jt}(G_T) - \{u_{ir}, u_{jt}\},$$

are possibly interfered by e. In other words, e' can be interfered by e only if e' is adjacent to some interface

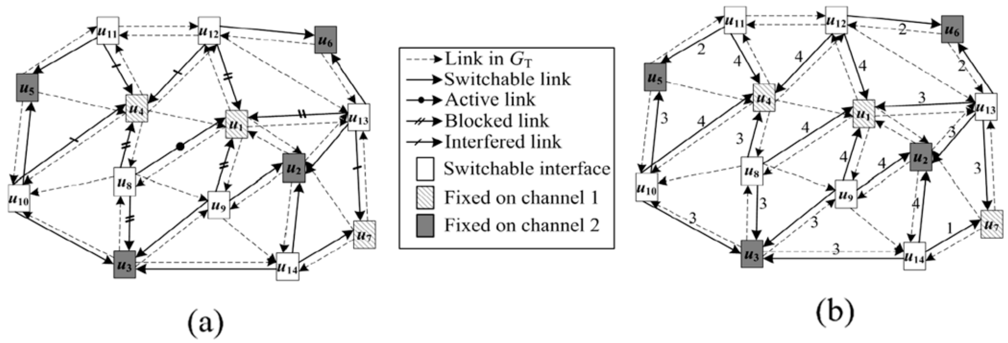

Fig. 3. (a) Blocked and interfered links of $u_8\,u_1$ and (b) $|IL(e)|$ of each $e \in E_S$.

in $PIN(e)$. With this fact, the set of links that are possibly interfered by e can be defined as

$$PIL(e) = \{e' | e' \in E_{ls}(G_S), u_{ls} \in PIN(e)\} - BL(e).$$

Notice that $BL(e)$ is subtracted from $PIL(e)$, since for any blocked link $e' \in BL(e)$, it can never be active simultaneously with e even when not interfered by e. Let $\chi(e)$ denote the channel of e, i.e. the channel assigned on u_{jt}. The set of *interfered links* of e can be given by

$$IL(e) = \{e' | e' \in PIL(e), \chi(e') = \chi(e)\}.$$

As shown in Fig. 3a, the interfered links of u_8u_1 are $u_{10}u_4, u_{11}u_4, u_{12}u_4$, and $u_{13}u_7$. Combining the two circumstances, we say that e' is the *conflicted link* of e whenever it is either blocked or interfered by e. The set of conflicted links of e is defined by

$$CL(e) = BL(e) \cup IL(e).$$

Our ultimate goal is to minimize the *total number of conflicted links* (*TCL*), i.e.

$$TCL = \sum_{e \in E_S} |CL(e)|.$$

Note that since the set of blocked links of any $e \in E_S$ is independent to the assignment of channels on V_F, the minimization of the *TCL* is equivalent to minimizing the *total number of interfered links* (*TIL*), i.e.

$$TIL = \sum_{e \in E_S} |IL(e)|. \tag{1}$$

In addition, for any given ρ, the cardinality of E_S is a constant. Hence, it is also equivalent to minimizing the average number of links that can be interfered as one of the switchable links is active. For example, in Fig. 3b, the value indicated on each link e is $|IL(e)|$. By summing up these values, we get $TIL = 66$. On the other hand, there are totally 21 switchable links. So, on average a transmission in this graph interfere with at most 3.14 links ($66/21 \cong 3.14$). That is, the *TIL* can be treated as an *average-case performance measurement at the link-layer*. The problem is now defined as follows:

Definition 2 (*Min TIL*). Given a transmission graph $G_T = (V_I, E_T \cup E'_T)$, a role assignment $\rho = (V_F, V_S)$, and H channels, find a channel assignment $\chi : V_F \to [H]$ such that the *TIL* is minimized.

In Section 6.1, we will show that minimizing the *TIL* is beneficial to improve network throughput by simulations.

4. Role assignment

This section begins by showing the $\mathcal{NP}$-hardness of the Min *TSL*. Next, we present a linear time 1/2-approximation algorithm to find nearly optimal solutions.

4.1. $\mathcal{NP}$-Hardness

Since the role assignment problem is first defined in this paper, we need to analyze its time complexity at the very beginning. Let us consider a special case of inputs, where the neighbor sets of all interfaces on the same node in G_C are equal, called the *identical* G_C. When each node has only two interfaces, the Max *TSL* with identical G_C is clearly trivial, since there is no choice but assigning one interface to be fixed and another to be switchable for each node. However, if a lot of nodes have more than two interfaces, the problem will become much complicated.

In the following theorem, we show that the Max *TSL* with more than two interfaces is $\mathcal{NP}$-hard even if the given G_C is identical. The proof is based on a polynomial time reduction from the *maximum cut problem* (Max Cut). The problem, as defined below, is one of the Karp's original $\mathcal{NP}$-complete problems [20].

Definition 3 [20] Max Cut. Given a graph $G = (V, E)$, find a partition $(S, \overline{S})$ of V such that the cardinality of $(S, \overline{S})$, i.e. the number of edges with one end point in S and another end point in $\overline{S}$, is maximized.

Theorem 1. *If the number of nodes having more than two interfaces is not a constant, the* Max *TSL with identical G_C is $\mathcal{NP}$-hard.*

The proof of Theorem 1 is given in Appendix A. The theorem indicates that the Max *TSL* cannot be optimally solved in polynomial time unless $\mathcal{NP} = \mathcal{P}$. Therefore, in the next subsection, we present an 1/2-approximate algorithm to find nearly optimal solutions.

4.2. 1/2-Approximate algorithm

The following design is based on an algorithm for the Max Cut proposed in [21]. Our algorithm is initiated by placing two arbitrary interfaces (here we take u_{i1} and u_{i2}) of each mesh node u_i into V_F and V_S, respectively. It is to ensure the network connectivity. Next, for each unconsidered interface u_{ir}, if u_{ir}'s neighbors in G_C that have been placed in V_F are fewer than those being placed in V_S, i.e. $|N_{ir}(G_C) \cap V_F| < |N_{ir}(G_C) \cap V_S|, u_{ir}$ is assigned to V_F; otherwise, it is assigned to V_S. In other words, an interface will be assigned to the set where the assigned interfaces has fewer linkages to itself. For example, in Fig. 4, interface u_{ir} will be assigned to V_F, since $|N_{ir}(G_C) \cap V_F| = 2 < |N_{ir}(G_C) \cap V_S| = 3$, where $N_{ir}(G_C) = \{a, b, \ldots, g\}$. On the other hand, the objective value *TSL* is increased by the number of links between u_{ir}'s and all u_{ir}'s neighbors that have been placed to different side. The algorithm is summarized below.

Algorithm 1. MAXTSL

Input: An identical communication graph $G_C = (V_I, E_C)$.
Output: A role assignment ρ and the *TSL*.
Step 1: $TSL := 0; V_F := \emptyset; V_S := \emptyset;$
Step 2: For each node $u_i \in V_N$,

$$V_F := V_F + \{u_{i1}\};$$
$$V_S := V_S + \{u_{i2}\};$$
$$TSL := TSL + |N_{i1}(G_C) \cap V_S| + |N_{i2}(G_C) \cap V_F|;$$

Step 3: For each interface $u_{ir} \in V_I - V_F - V_S$,
if $|N_{ir}(G_C) \cap V_F| < |N_{ir}(G_C) \cap V_S|$,

$$V_F := V_F + \{u_{ir}\};$$
$$TSL := TSL + |N_{ir}(G_C) \cap V_S|;$$

otherwise,

$$V_S := V_S + \{u_{ir}\};$$
$$TSL := TSL + |N_{ir}(G_C) \cap V_F|;$$

Step 4: Stop, and return $\rho = \{V_F, V_S\}$ and *TSL*.

The following theorem shows that any *TSL* resulted from this algorithm is no less than a half of the optimal value.

Theorem 2. *Algorithm MAXTSL is a 1/2-approximate algorithm for the* MAX *TSL with identical* G_C.

The proof of Theorem 2 is given in Appendix A. The time complexity of the Algorithm MAX *TSL* is analyzed as follows. Note that each interface is examined exactly once by either Step 2 or Step 3. Moreover, for each u_{ir}, the calculation of $|N_{ir}(G_C) \cap V_F|$ and $|N_{ir}(G_C) \cap V_S|$ needs to check at most $|N_{ir}(G_C)|$ links. Hence, the time complexity is *linear* to the total number of links in G_C, i.e. $O(|E_C|)$.

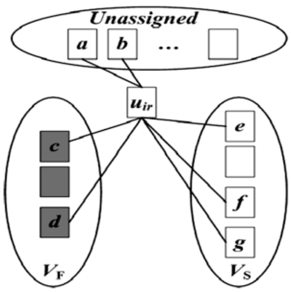

Fig. 4. An example of ALGORITHM MAXTSL.

For the general G_C, the MAX *TSL* would become more complicated because finding a feasible assignment is no longer trivial. Consider an example in Fig. 5, where the G_C is not identical. The two assignments in Fig. 5b and c have been specified a pair of fixed and switchable interfaces to each node. But, the resulting graph in Fig. 5c is disconnected. It is reasonable to assert that the problem is $\mathcal{NP}$-hard even if the goal is simply to find a feasible assignment. If our assertion is true, designing a branch-and-bound algorithm that enumerates a confined solution space could be more appropriated. The proof of this assertion is part of our ongoing work.

5. Semi-fixed channel assignment

In this section, we investigate into the channel assignment problem defined in Section 3.3. The $\mathcal{NP}$-hardness of the MIN *TIL* is obvious. Consider the case of an identical G_C with two interfaces on each node. It is equivalent to the traditional receiver-based channel assignment problem with single interface [12]. Therefore, we will focus here on designing an efficient algorithm for the MIN *TIL*.

The main idea of this algorithm is based on problem transformation. It consists of three parts: First, we transform our problem into a coloring-based problem, called the *minimum k-partition problem* (MIN *K*-PARTITION). Next, a greedy algorithm is designed to find a sufficiently good solution σ' for the MIN *K*-PARTITION. Finally, the solution σ' will be converted into a channel assignment χ with the same objective value for our problem. Now, the three parts are given in the following subsections.

5.1. Problem transformation

The MIN *K*-PARTITION [22] is a coloring-based problem. Its purpose is to find a set of edges with minimum total weight whose removal makes a graph *K*-colorable. The formal definition is as follows.

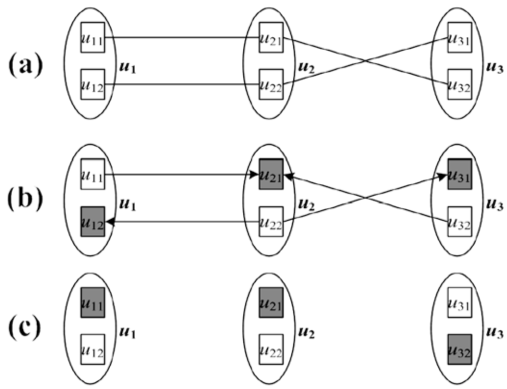

Fig. 5. (a) General G_C; (b) feasible assignment and (c) infeasible assignment.

Definition 4 [22]. Given a graph $G = (V, E)$, with a weight function $w : E \to N$, find a K-color assignment $\sigma : V \to [K]$ such that the total weight W of monochromatic edges is minimized, where

$$W = \sum_{v_i v_j \in E : \sigma(v_i) = \sigma(v_j)} w(v_i, v_j). \tag{2}$$

In the following context, we transform any input X of the Min *TIL* into an input Y of the Min K-Partition. It can be shown that the optimization of our problem to X is equivalent to solving the Min K-Partition to Y. First of all, from Eq. (1), we have the following derivation for the *TIL*:

$$\begin{aligned} TIL &= \sum_{e \in E_S} |IL(e)| = \sum_{u_{ir} \in V_F} \sum_{e \in E^+_{ir}(G_S)} \left| \bigcup_{u_{jt} \in V_F} \{e' \in E^+_{jt}(G_S) | e' \in IL(e)\} \right| \\ &= \sum_{u_{ir} \in V_F} \sum_{e \in E^+_{ir}(G_S)} \left| \bigcup_{u_{jt} \in V_F} \{e' \in PIL(e) \cap E^+_{jt}(G_S) | \chi(e') = \chi(e)\} \right| \\ &= \sum_{(u_{ir}, u_{jt}) \in V_F \times V_F} \sum_{e \in E^+_{ir}(G_S)} \left| \{e' \in PIL(e) \cap E^+_{jt}(G_S) | \chi(e') = \chi(e)\} \right|. \end{aligned}$$

Let

$$\tau(u_{ir}, u_{jt}) = \sum_{e \in E^+_{ir}(G_S)} \left| \{PIL(e) \cap E^+_{jt}(G_S)\} \right|. \tag{3}$$

The *TIL* can be represented as

$$TIL = \sum_{(u_{ir}, u_{jt}) \in V_F \times V_F : \chi(u_{ir}) = \chi(u_{jt})} \tau(u_{ir}, u_{jt}), \tag{4}$$

where $\chi(u_{ir})$ and $\chi(u_{jt})$ are channels assigned on u_{ir} and u_{jt}. Eq. (4) indicates that for any fixed interface u_{jt}, if there is an interface u_{ir} fixed on the same channel (i.e. $\chi(u_{ir}) = \chi(u_{jt})$), the total number of links incident to u_{jt} that will be interfered by any transmission to u_{ir} is $\tau(u_{ir}, u_{jt})$. In other words, the increment to the *TIL* caused by assigning u_{ir} and u_{jt} on the same channel is the sum of $\tau(u_{ir}, u_{jt})$ and $\tau(u_{jt}, u_{ir})$.

Based on this relation, a transformation from the Min *TIL* to Min K-Partition is defined as follows.

Definition 5 (*Transformation*). Given a transmission graph $G_T = (V_I, E_T \cup E'_T)$, a role assignment $\rho = (V_F, V_S)$, and H channels for the Min *TIL*, we construct an instance, including a graph $G' = (V', E')$, a weight function w', and a constant K', for the Min K-Partition such that

(i) $V' = \{v_{ir} | u_{ir} \in V_F\}$;
(ii) $E' = \{v_{ir} v_{jt} | u_{ir} \in V_F, u_{jt} \in V_F\}$;
(iii) $w'(v_{ir}, v_{jt}) = \tau(u_{ir}, u_{jt}) + \tau(u_{jt}, u_{ir}), \forall v_{ir} v_{jt} \in E'$;
(iv) $K' = H$.

Now, we show their equivalence. Let $opt_{\mathscr{P}}(I)$ denote the optimal value of a problem $\mathscr{P}$ with an input instance I. We have the following result.

Theorem 3. *Given a transmission graph $G_T = (V_I, E_T \cup E'_T)$, a role assignment $\rho = (V_F, V_S)$, and H channels,*

$$opt_{\text{Min } TIL}(G_T, \rho, H) = opt_{\text{Min } K\text{-Partition}}(G', w', K').$$

The proof of Theorem 3 is also given in Appendix A. Theorem 3 indicates that with the transformation in Definition 5, any existing algorithm for the Min K-Partition can be applied to solve our problem and obtain the same objective value. An example is shown in Fig. 6a, where the weighted graph is transformed from the transmission graph in Fig. 3a. The weight $w(v_{ir}, v_{jt})$ on each edge $v_{ir} v_{jt}$ is calculated from Table 1. The table shows the $\tau(u_{ir}, u_{jt})$ (marked in boldface) for any pair of fixed interfaces u_{ir} and u_{jt} in Fig. 3a. For instance, $w(v_1, v_4) = \tau(u_1, u_4) + \tau(u_4, u_1) = 10 + 14 = 24$. Note that any edge of zero weight (e.g. $v_3 v_6$) can be removed without loss of generality. A 2-color assignment of the weighted graph is given in Fig. 6b. We can see that the total weight $W = 66$, which is exactly equal to the *TIL* of Fig. 3b, obtained before.

A number of studies have been conducted for the min K-Partition. For $K = 2$, Garg et al. [22] presented an $O(\log |V|)$-approximation based on the multi-commodity flow technique. For $K = 3$, it can be approximated in $\varepsilon |V|^2$ for any $\varepsilon > 0$ [23]. However, for $K > 3$, it has been shown that the problem cannot be approximated in $O(|V|^{2-\varepsilon})$ for any $\epsilon > 0$ [23]. In other words, there is no constant-ratio approximation algorithm for the Min K-Partition unless $\mathscr{NP} = \mathscr{P}$. Therefore, in the next subsection, we intend to design a greedy algorithm to find "sufficiently good" solution for the Min K-Partition.

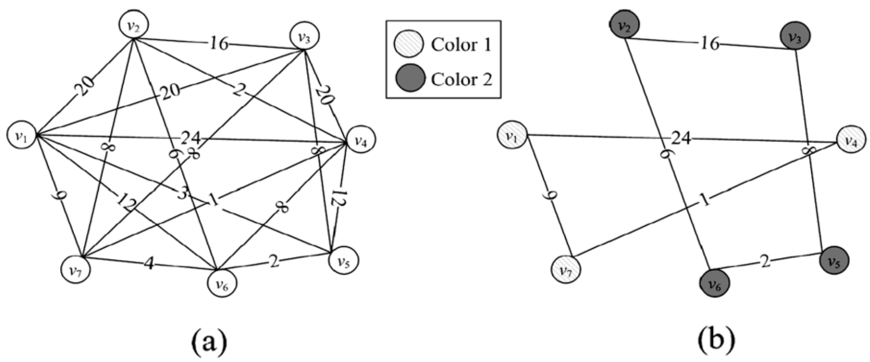

Fig. 6. (a) Weighted graph transformed from Fig. 3 and (b) a 2-color assignment.

Table 1
Derivation of Eq. (3) from Fig. 3.

u_{ir}	$e \in E^{+}_{ir}(G_S)$	u_{jt}							
		u_1	u_2	u_3	u_4	u_5	u_6	u_7	$\lvert PIL(e)\rvert$
u_1	u_8u_1	0	3	3	3	1	2	1	13
	u_9u_1		2	3	2	0	2	2	11
	$u_{12}u_1$		3	2	3	1	1	1	11
	$u_{13}u_1$		2	2	2	0	1	1	8
	$\tau(u_1, u_{jt})$		**10**	**10**	**10**	**2**	**6**	**5**	
u_2	u_9u_2	3	0	3	0	0	1	2	9
	$u_{13}u_2$	3		2	0	0	1	1	7
	$u_{14}u_2$	4		3	0	0	1	1	9
	$\tau(u_2, u_{jt})$	**10**		**8**	**0**	**0**	**3**	**4**	
u_3	u_8u_3	3	2	0	3	1	0	1	10
	u_9u_3	3	2		2	1		1	9
	$u_{10}u_3$	2	2		3	1	0	1	9
	$u_{14}u_3$	2	2		2	1	0	1	8
	$\tau(u_3, u_{jt})$	**10**	**8**		**10**	**4**	**0**	**4**	
u_4	u_8u_4	3	1	3	0	2	1	0	10
	$u_{10}u_4$	4	0	3		1	1	0	9
	$u_{11}u_4$	4	0	2		1	1	0	8
	$u_{12}u_4$	3	1	2		2	1	1	10
	$\tau(u_4, u_{jt})$	**14**	**2**	**10**		**6**	**4**	**1**	
u_5	$u_{10}u_5$	0	0	3	3	0	0	0	6
	$u_{11}u_5$	1	0	1	3		1	0	6
	$\tau(u_5, u_{jt})$	**1**	**0**	**4**	**6**		**1**	**0**	
u_6	$u_{12}u_6$	3	1	0	3	1	0	1	9
	$u_{13}u_6$	3	2	0	1	0		1	7
	$\tau(u_6, u_{jt})$	**6**	**3**	**0**	**4**	**1**		**2**	
u_7	$u_{13}u_7$	3	2	1	0	0	1	0	7
	$u_{14}u_7$	1	2	3	0	0	1		7
	$\tau(u_7, u_{jt})$	**4**	**4**	**4**	**0**	**0**	**2**		

5.2. Algorithm for the MIN K-PARTITION

The algorithm is presented as follows. At the beginning, the total weigh W is set as 0. Besides, we initiate an empty set S to keep track of any vertex that has been assigned a color. Before describing the following processes, let us first make some observations on the relation between W and S. Assume that a subset S of vertices in V have been assigned colors and the W has been updated to the total weight of monochromatic edges among vertices in S. Now, if we want to assign a color k to an unassigned vertex $v_i \in V - S$, the W can be increased by

$$W^{+}(v_i, S, k) = \sum_{v_j \in \{N_i(G) \cap S\}: \sigma(v_j) = k} w(v_i, v_j), \tag{5}$$

since $W^{+}(v_i, S, k)$ is the total weight of edges connecting v_i and its neighbors that have been assigned the same color. On the other had, for any v_i's neighbor v_j in S, if $\sigma(v_j) \neq k$, the weight $w(v_i, v_j)$ will not be counted in W. Hence, the upper bound of the W can be decreased by

$$W^{-}(v_i, S, k) = \sum_{v_j \in \{N_i(G) \cap S\}: \sigma(v_j) \neq k} w(v_i, v_j). \tag{6}$$

Combining Eqs. (5) and (6), we can observe that the final value of the W can be reduced by finding a smaller $W^{+}(v_i, S, k)$ and/or a larger $W^{-}(v_i, S, k)$. Hence, in our algorithm, we prefer to give higher priority to assigning a color k to an unassigned vertex v_i such that

$$W^{+}(v_i, S, k) - W^{-}(v_i, S, k) \tag{7}$$

is minimal. After assigning v_i, the S and W should be updated so that $S = S + \{v_i\}$ and $W = W + W^{+}(v_i, S, k)$. The above processes will continue until all vertices are assigned. The algorithm is summarized below.

Algorithm 2. MINKPT

Input: A graph $G = (V, E)$, a weight function $w : E \rightarrow \mathbf{N}$, and K colors.
Output: A *K-color* assignment σ and the total weight W.
Step 1: $W := 0; S := \emptyset$;
Step 2: Choice a vertex $v_j \in V - S$ and a color $k \in [1, \ldots, K]$ such that $W^{+}(v_i, S, k) - W^{-}(v_i, S, k)$ is minimal;
Step 3: Assign $\sigma(v_j) = k$;
Step 4: $W := W + W(v_i, k); S := S + \{v_i\}$;
Step 5: If $S \neq V$, go back to Step 2;
Step 6: Stop, and return σ and W.

5.3. Algorithm for the MIN TIL

Based on the previous two subsections, a channel assignment algorithm for the MIN *TIL* is now presented below. The input is first transformed into an instance of the MIN *K*-PARTITION, according to Definition 5. Next, the MIN KPT is applied to solve the transformed instance and obtain a color assignment σ' and the corresponding total weight

W'. The σ' and W' are then converted into a channel assignment χ and the objective value *TIL*, according to Eqs. (8) and (9) in the proof of Theorem 3 at Appendix A.

Algorithm 3. MINTIL

Input: *A transmission graph* $G_T = (V_I, E_T \cup E'_T)$, *a role assignment* $\rho = \{V_F, V_S\}$, *and H channels.*
Output: A channel assignment χ *and the TIL.*
Step 1: Transform G_T, ρ, *and H into* G', w' *and* K' *according to Definition 5.*
Step 2: Apply the MINKPT to $G' = (V', E')$ *with* w' *to find a* K'*-color assignment* σ' *and the total weight* W'.
Step 3: For any $u_{ir} \in V_F$, *assign* $\chi(u_{ir}) = \sigma'(v_{ir})$.
Step 4: Stop, and return χ *and* $TIL = W'$.

The time complexity of the MINTIL is analyzed in the following: About the transformation in Step 1, the input consists of $|V_F|^2$ pairs of fixed interfaces. For each pair of u_{ir} and u_{jt}, the calculation of $\tau(v_{ir}, v_{jt})$ and $\tau(v_{jt}, v_{ir})$ needs $O(|V_S|^2)$ time, since there are at most $|V_S|$ edges in $E^+_{ir}(G_S)$ and $E^+_{ir}(G_S)$. Hence, Step 1 can be done in $O(|V_F|^2|V_S|^2)$. About the MINKPT in Step 2, for each iteration of assigning a color to an unassigned vertex, there are $O(K'|V'|)$ choices. For a choice, the calculation of Eq. (7) takes $O(|V'|)$ time. Moreover, there are totally $|V'|$ iterations. Besides, we know that $|V'| = |V_F|$ and $K' = H$. Thus, Step 2 can be done in $O(H|V_F|^3)$. Combining together, the time complexity of the MINTIL is $O\left(\max\{|V_F|^2|V_S|^2, H|V_F|^3\}\right)$.

6. Experiment results

In this section, we conduct simulations using the ns2 simulator. Our experiment consists of three parts: In the first part, we provide statistical results to show that optimizing MINTIL and MAXTSL indeed help to improve the performance. Next, we compare the single-hop and multi-hop performances of our algorithm with other approaches in the last two parts.

The network environment and test scenarios are mostly adapted from those in [26], in which a flow-independent protocol (CLICA) similar to our approach was proposed. For any network under test, 50 static mesh nodes are randomly deployed on a 1000 m by 1000 m area. Each mesh node has an equal number of radio interfaces (R) with an identical transmission range of 250 m and interference range of 550 m. The number of channels (H) will be varied from 3 to 12. The data rate is fixed on 2 Mbps. Any channel switch will incur a hardware delay of 1 ms. Note that although higher data rates are specified in IEEE 802.11 standards (e.g. 802.11 b/g) we are more interested in relative performance behavior.

In order to spread traffic load equally onto different channels, each node maintains a separate queue for any channel it can use. When some packet arrived at a sender, the packet will be dispatched to the queue having least packets among all channels which are communicable to the intended receiver. For example, if node A received a packet destined for node B and found that it can communicate with B using ch_1 and ch_2, the packet will be dispatched to the queue for ch_1 as long as existing packets in this queue is less than that for ch_2. Besides, if a node can communicate with its neighbor via multiple interfaces, the traffic (packets) will be randomly striped across interfaces for parallel transmissions. These two mechanisms will be applied to all protocols in our simulation.

6.1. Problem evaluations

First, we evaluate our channel assignment problem, i.e. Min *TIL*, by computing the correlation coefficient between the *TIL* and the aggregated throughput over $H = 3, 7, 12$ and $R = 2, 3, 4$. For each combination of H and R, we generate 20 random networks, i.e. the transmission graph G_T. The role assignment of each G_T is specified using a simple random algorithm (RAN), where each mesh node is designated one pair of switchable and fixed interfaces to ensure the network connectivity and the remaining interfaces are assigned at random. Next, for each network, we systematically generate 100 various channel assignments from the solution space, such that the ith assignment has at least $i\%$ of fixed interfaces on the same channel and the others are arbitrary. This is to ensure that results are obtained from a wide variety of channel assignments with different quality. On top of each network, we establish unicast flows (back and forth) with identical poisson packet arrivals between every pair of neighboring nodes in the network. The mean packet arrival rate is 0.5 Mbps, which is large enough to achieve the saturated throughput. The packet size is 1024 bytes. Each simulation will last for 50 s.

The correlation coefficients averaged from the 20 networks for each pair of (R, H) are summarized in Table 2. From Table 2, we learn that the correlation coefficient shows a strong negative association between the *TIL* and the aggregated throughput. That is, the smaller *TIL* the network has, the higher throughput it will achieve. In particular, when the numbers of channels and interfaces are large, the value is very close to -1. For example, when $R = 4$ and $H = 12$, the correlation coefficient is -0.963. This means that the *TIL* and the aggregated throughput are almost linearly correlated. There are two reasons: First, for a given H, the co-channel interference among interfaces will become more serious if there are more interfaces on nodes. According to Eq. (4), the *TIL* is the total number of links incident to any pair of fixed interfaces which are on the same channel. Thus, the *TIL* can faithfully reflect the increasing interference. Second, a larger value of R and/or H corresponds to a larger solution space of possible channel assignments, which will enlarge the gap between the worst and optimal results of the *TIL*.

Next, we evaluate the role assignment problem, i.e. Max *TSL*. As discussed in Section 3.2, a larger *TSL* implies more possible switching pairs between interfaces, but it does

Table 2
Correlation coefficients between *TIL* and throughput.

	$H = 3$	$H = 7$	$H = 12$
$R = 2$	−0.884	−0.889	−0.908
$R = 3$	−0.914	−0.928	−0.937
$R = 4$	−0.928	−0.948	−0.963

not imply that nodes can switch their switchable interfaces to non-interfered channel more easily, unless channels are properly arranged among fixed interfaces. The *TIL* has been shown a suitable cost metric to find proper channel assignment. Therefore, instead of showing the correlation coefficient between the *TSL* and the aggregated throughput with random channel assignments, we evaluate the Max *TSL* problem by directly comparing the aggregated throughputs resulted from the two combinations of RAN + MINTIL and MAXTSL + MINTIL (Recall that RAN is the random role assignment algorithm described above). Fig. 7a and b report the results under varied numbers of channels and interfaces. In the two subfigures, we can see that MAXTSL shows clear gain over the random algorithm under the same channel assignment. The improvement can be more significant as R and H are large, since the more channels (interfaces) the network has, the more possible channel switching nodes can do. In summary, the above results have confirmed that minimizing the *TIL* and maximizing the *TSL* are indeed beneficial to improve the performance.

6.2. Comparisons: Single-hop performance

Now, we compare the MAXTSL + MINTIL with the following channel assignment protocols (algorithms) to study the relative performance:

1. Hybrid multi-channel protocol (HMCP) [16,17]: The HMCP was the only channel assignment approach designed for the hybrid strategy so far. Recall that this protocol requires each node to periodically check its channel statuses and broadcast a Hello message on every channel whenever some fixed channel was changed. We set the checking period as 5 s, which is the default value in [16]. In addition, to compare HMCP with our channel assignment algorithm on the same base, the roles of interfaces for HMCP are also determined by MAXTSL (i.e. MAXTSL + HMCP).
2. Common channel assignment (CCA)[25]: Akin to our design, the CCA is a traffic-independent approach. It assigns a common set of channels to all nodes. More precisely, the rth interface at each node i, i.e. u_{ir}, is assigned the rth channel.
3. Connected-low-interference channel assignment (CLICA) [26]: The CLICA is also independent to any specific traffic pattern. Besides, similar to our approach, it allows each node to operate on a different set of channels and is based on the greedy method. A phase-2 procedure in [26] that assigns channels to uncolored radio interfaces has been implemented here.

Other details about these protocols have been reviewed in Section 2. In addition, the performance in single channel case serves as a baseline in our comparison.

Between every pair of neighboring nodes in the network, we establish two poisson flows (back and forth) with mean packet arrival rate of $x/2L$ Mbps, where L is the total number of neighboring node pairs and x is the expected load offered to nodes. We will compare the link-layer performance under different x started from 2 to 30 Mbps with an increment of 2 Mbps. Any result point is averaged from 20 networks and the simulation of each network lasts for 50 s.

Fig. 8 reports the aggregated throughput for four combinations of R and H. The cases for 3 and 12 channels are representative of 802.11b and 802.11a networks, respectively. As shown in Fig. 8a, when $H = 3$, the CLICA, MAXTSL + HMCP and MAXTSL + MINTIL achieve very similar performance, because it is very easy to fully exploit the channel bandwidth if providing only a small number of channels. Even so, the MAXTSL + MINTIL obtains the largest throughput in this case.

When H is large, as shown in Fig. 8b–d, our approach can perform significantly better than other approaches, especially when nodes are loaded with higher packet rates. Under 30 Mbps offered load, the MAXTSL + MINTIL

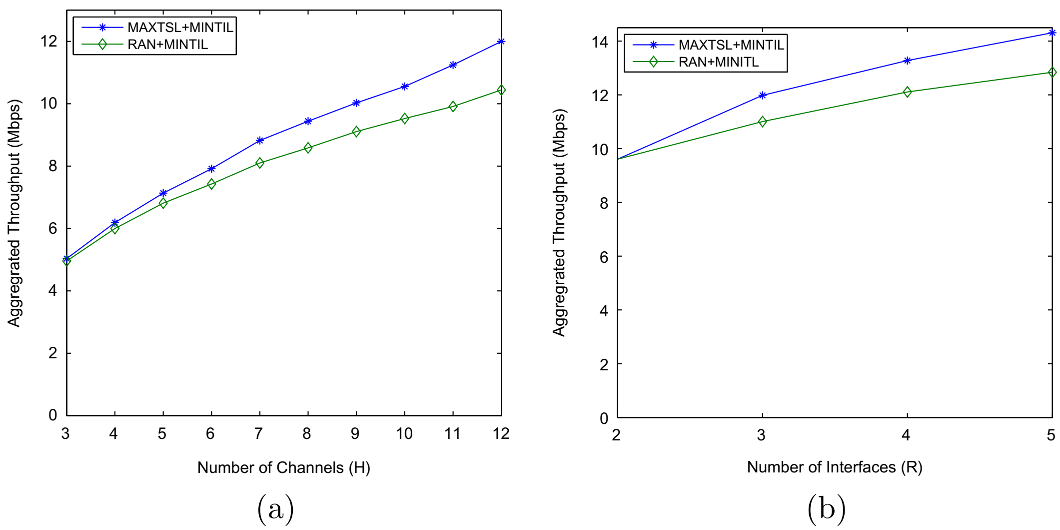

Fig. 7. Aggregated throughput (a) $R = 5$ and (b) $H = 12$.

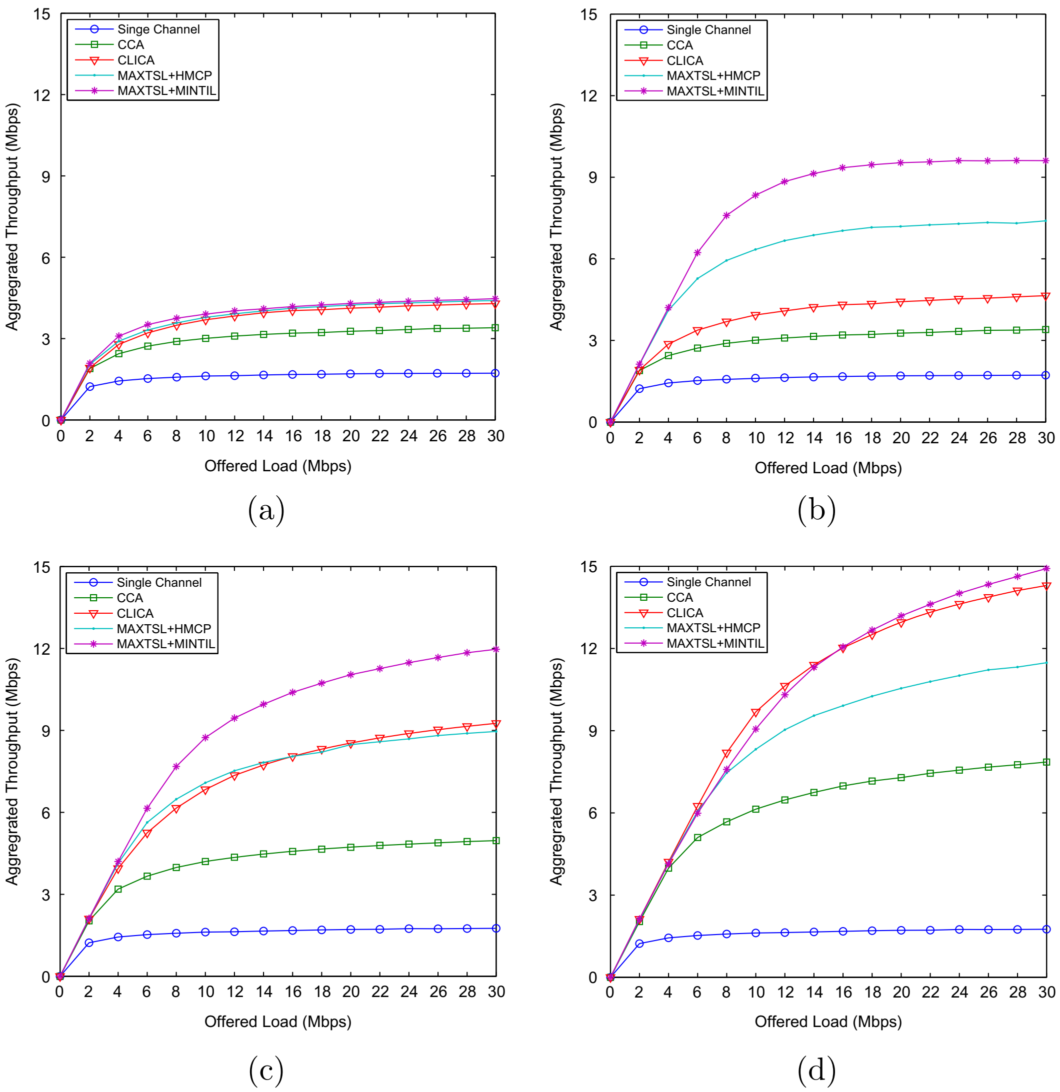

Fig. 8. Aggregated throughput (a) $(R = 2, H = 3)$; (b) $(R = 2, H = 12)$; (c) $(R = 3, H = 12)$ and (d) $(R = 5, H = 12)$.

improves the throughput in single channel case up to a factor of 5.58 with 2 interfaces, a factor of 6.83 with 3 interfaces, and a factor of 8.51 with 5 interfaces. This result indicates that MAXTSL + MINTIL can effectively exploit multiple channels by using only a small number of interfaces. There are two main sources of such impressive performance. The first stems from the genius of our concerned strategy. The hybrid strategy allows each node to utilize diverse channels via its switchable interface(s). Thus, nodes are more likely to find non-interfered channels. The second is due to the superiority of our designed algorithms. The MAXTSL and MINTIL can produce more non-interfered switching pairs so as to further reduce interference.

By contrast, the performance of CCA is strictly limited by the number of interfaces (R) on each node so that only R channels can be utilized at anytime. The CLICA allows nodes to operate on different sets of channels. Hence, it can provide much more throughput than CCA. However, since CLICA is a static approach, the number of channels that can be used by each node is still limited by R. As shown in Fig. 8b, with 2 interfaces and 12 channels, it only achieves at most 2.4 times throughput compared with the single channel case (Recall that ours can achieve 5.58 times in this case). For a comparable improvement (see Fig. 8d), the CLICA requires at least 5 interfaces on each node, which is very cost-inefficient in practice.

Similar to our approach, the MAXTSL + HMCP also follows the hybrid strategy. Nonetheless, its throughput is not comparable to ours in particular when H is large. As shown in Fig. 8b–d, it only achieves less than 80% of our throughput when $H = 12$ and $x = 30$ Mbps. There are three possible reasons: (1) Although HMCP can uniformly assign channels to fixed interfaces according to the channel's usage on neighboring nodes, it does not take the possible channel switching into consideration. As a result, two potentially interfering fixed interfaces could be assigned

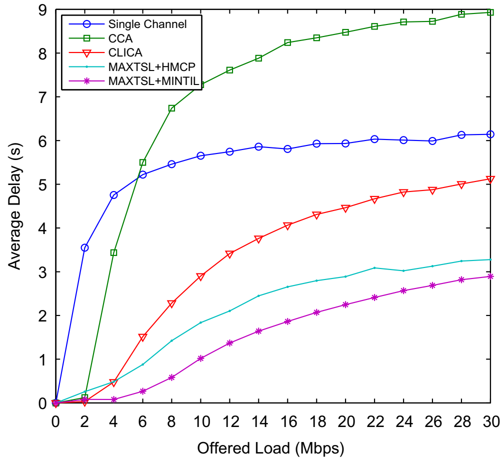

Fig. 9. Average delay with $R = 3$ and $H = 12$.

the same channel even if there are many switchable interfaces possibly switching to them; (2) Each node in this approach has to broadcast a Hello message on every channel whenever some channel is changed at the periodic check point. The broadcasting process may consume considerable bandwidth; (3) A miss matching channel switch may occur when some updated channel was not received, incurring additional cost for retransmissions (see Section 1 for more detail explanation).

Fig. 9 shows the average delay with 3 interfaces and 12 channels. (Note that we concentrate on the case of $R = 3$ and $H = 12$ for our hereafter comparisons). We can see that MAXTSL + MINTIL has the lowest delay compared with other approaches. Interestingly, although MAXTSL + HMCP and CLICA has almost the same performance under this setting (see Fig. 8c), the delay of MAXTSL + HMCP is much lower than that of CLICA. This phenomenon is explained as follows: In the hybrid strategy, each node can forward packets using diverse channels to its neighboring nodes. Therefore, a node has to maintain more packet queues for various channels. This means that the average queue length of each channel is shorter than that of CLICA. As a result, each packet will stay in queue for a shorter period of time before being sent out. This is also the reason that MAXTSL + MINTIL has lower delay. Another reason is that MAXTSL + MINTIL can greatly reduce interference, in turn leading to lower channel access delays from the MAC layer (include backoffs and retransmission). Notice that although CCA has larger throughput than the single channel case its delay rises drastically as the offered load over a certain limit. It is possibly due to the fact that the number of interfaces assigned to each channel is still equal to that of the single channel case, resulting in the same level of interference within each channel.

In Fig. 10a, we compare the performance under 50 nodes and 100 nodes for each protocol. The performance ratio (PR_1) is obtained by dividing the aggregated throughput of 100 nodes by that of 50 nodes. Observe that the performance ratio (PR_1) of any static approach (include CLICA, CCA, and single channel case) degrades as long as the offered load increases and below 1 when the load is over a certain limit. This phenomenon is possibly explained by the reason that interference in 100-node network is much severer than that of 50-node network but each node still utilizes a fixed number of channels. By contrast, the performance ratio of any hybrid approach (include MAXTSL + MINTIL and MAXTSL + HMCP) increases with an increment in the offered load. Therefore, the hybrid strategy is more suitable for a large scale network.

Fig. 10b reports a comparison accessing the distance between the protocol and the physical interference models. The channel access is resolved by the 802.11 DCF in the protocol model, while it only depends on the signal-to-noise ratio (SINR) in the physical model. The performance ratio (PR_2) is equal to the aggregated throughput of the physical model divided by that of the protocol

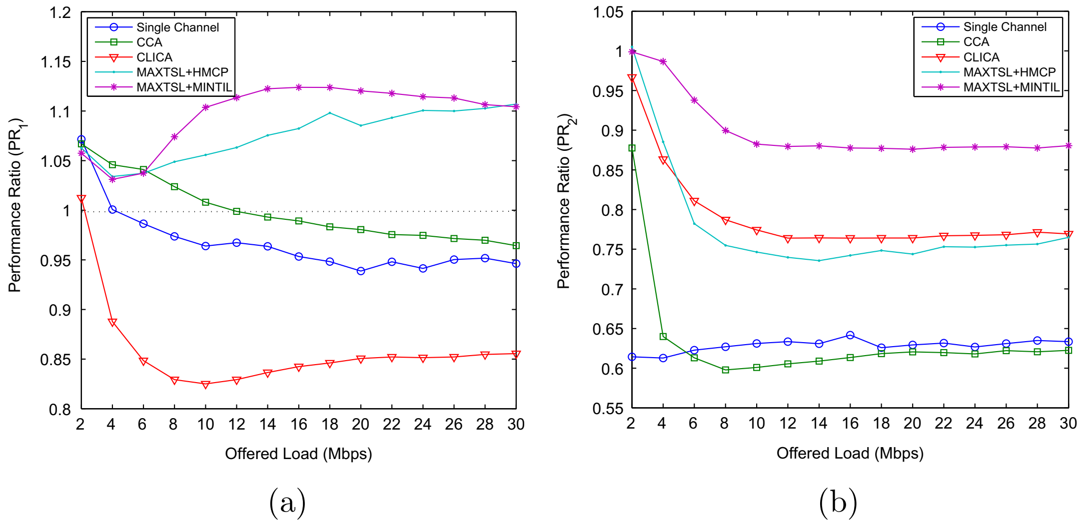

Fig. 10. Performance ratios vs. offered load with $R = 3$ and $H = 12$ (a) PR_1 = throughput under 100 nodes/throughput under 50 nodes and (b) PR_2 = throughput under physical model/throughput under protocol model.

model. We can see that the performance ratios (PR_2) of CCA and single channel case degrade over 35% in the physical model. The degradation is mainly caused by hidden terminal nodes in each channel. The CLICA, MAXTSL + HMCP, and MAXTSL + MINTIL can spread nodes (interfaces) on different channels. Thus, they can greatly mitigate the hidden terminal problem. As shown in Fig. 10b, there are relatively lower degradations in their performance ratios. Noticeably, although our approach is designed under the protocol model, it suffers only 12% decrement in throughput when applying to the physical model.

6.3. Comparisons: multi-hop performance

Lastly, we test the multi-hop performance. Following the settings in [26], we apply 50 s one-way bulk transfer with FTP application in our test. Two different traffic patterns are considered. For the Internet access pattern, four Internet gateway nodes are randomly chosen from the 50 nodes. Each non-gateway node has a data transfer to the nearest gateway node determined by the shortest path length in terms of hops. For the peer-to-peer traffic pattern, 100 source-destination pairs (chosen at random) separately start up a data transfer.

Fig. 11a and b show the average end-to-end TCP throughput for the two experiments. Normalized results (divided by the throughput in single channel case) are drawn in Fig. 11c and d. We can see that MAXTSL + MINTIL substantially improves single channel performance especially for large-hop flows. The normalized throughput of our approach is also greater than the others in both cases. These results reveal that MAXTSL + MINTIL is not only superior in reducing inter-flow interference but also

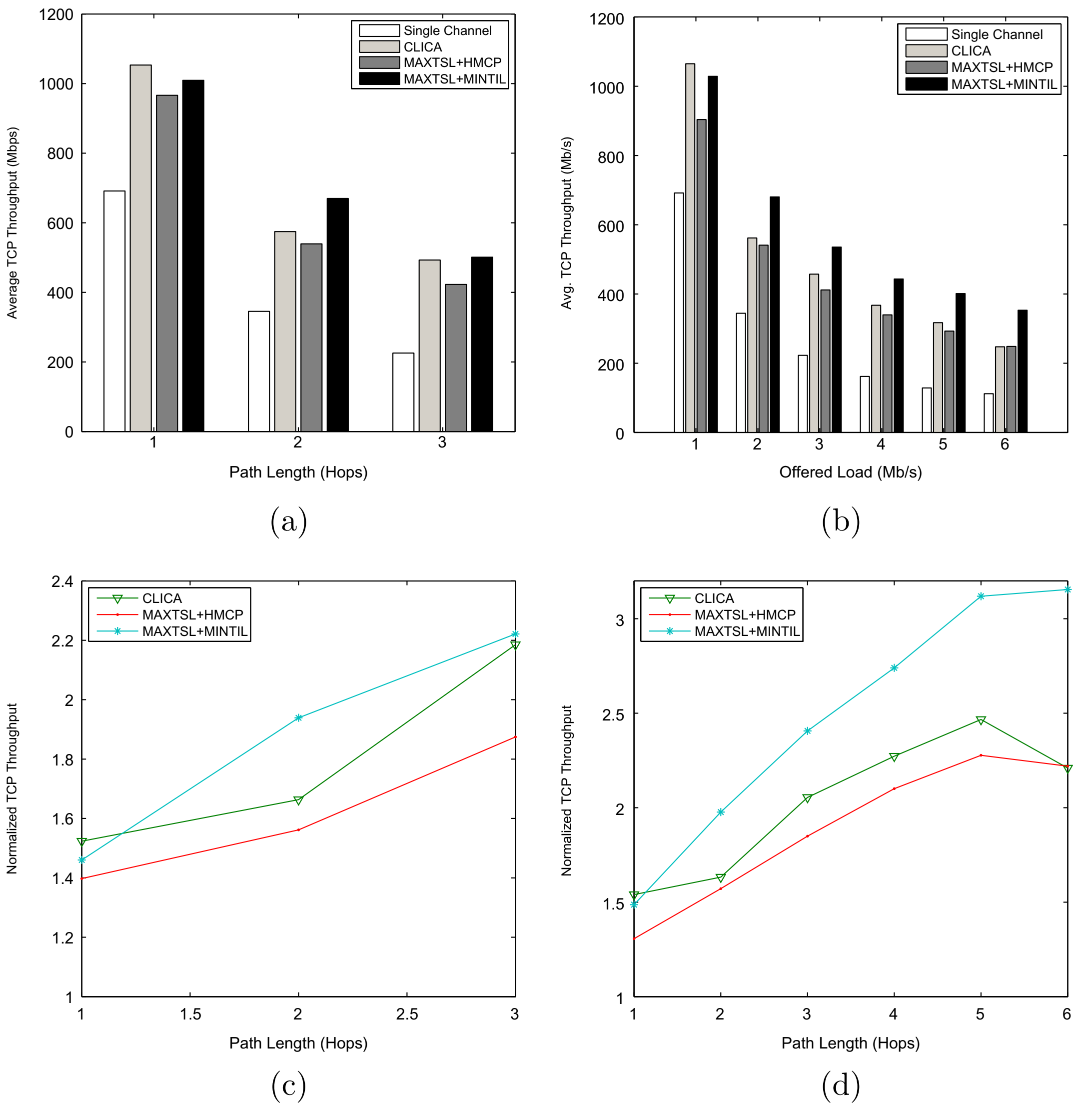

Fig. 11. TCP throughput vs. path length with $R = 3$ and $H = 12$ (a) Aggregated TCP throughput under Internet access pattern; (b) aggregated TCP throughput under peer-to-peer pattern; (c) normalized TCP throughput under Internet access pattern and (d) normalized TCP throughput under peer-to-peer pattern.

superior in mitigating inter-hop interference to each flow. Note that since the tested flows are started separately, there is neither inter-flow interference nor inter-hop interference during any 1-hop flow. Besides, any channel switching requires a hardware delay of 1 ms. Therefore, our approach is slightly inferior to CLICA in 1-hop performance.

7. Conclusion

In this paper, we have defined two optimization problems to characterize the unique feature of the hybrid strategy. The $\mathcal{NP}$-hardness of the two defined problems have been proved. In order to solve our problems in reasonable time, we have designed efficient algorithms. Simulated results have shown that optimizing our defined problems indeed help to improve the performance. Besides, our algorithms are significantly superior to existing hybrid channel assignment approach and flow-independent approaches.

For the future research, it is worthwhile to extend our centralized algorithms to distributed protocols. We believe that the extensions are possible. Besides, we can see that the *TIL* is partially a function of the topology in G_S which is further determined by the roles of interfaces. Therefore, to obtain the global optimum of the *TIL*, the two problems should be jointly considered. This is one of our ongoing works.

Acknowledgement

The authors would like to thank the anonymous referees for their valuable comments, as well as Chi-Yu Li and Chin-Ching Chen for their helpful assistance in conducting the experimental results.

Appendix A. Proofs

Proof of Theorem 1. Given two integers p and q, we denote $TSL(q)$ and CUT (p) as two decision problems that determine whether $|E_S| \geqslant p$ and $|(S,\overline{S})| \geqslant q$, respectively. We have the following argument.

Given a graph $G = (V,E)$, we constitute an instance $G_C = (V_I, E_C)$ for the $TSL(q)$: For any $v_i \in V$, there is a node u_i with three interfaces u_{i1}, u_{i2} and u_{i3} in V_I. In addition, a node u_0 with two interfaces u_{01} and u_{02} is in V_I. For any $v_iv_j \in E$, a link is between u_{ir} and u_{jt} in $E_C, \forall r,t = 1,2,3$. Besides, for any $v_i \in V$, a link is between u_{ir} and u_{0t} in $E_C, \forall r = 1,2,3$ and $t = 1,2$. We will show that G has a cut of cardinality no less than p if and only if there is a feasible assignment of G_C such that $|E_S| \geqslant q = 4|E| + 3|V| + p$.

Consider a cut $(S,\overline{S})$ of G. A partition $\rho = (V_F, V_S)$ is constructed by: (a) $u_{i1} \in V_S$ and $u_{i2} \in V_F, \forall v_i \in V$; (b) $u_{i3} \in V_S, \forall v_i \in S$; (c) $u_{i3} \in V_F, \forall v_i \in \overline{S}$. By (a), each node u_i has a switchable interface u_{i1} and a fixed interface u_{i2}. Besides, node u_0 has two links $u_{02}u_{i1}$ and $u_{01}u_{i2}$, respectively, from and to any u_i, i.e. u_0 is a center point connecting all nodes. Thus, the corresponding $\overline{G}_S$ is strongly connected. Now consider $|E_S|$. We decompose E_S into the following disjointed sets:

$$E_{S,1} = \{u_{i1}u_{j2}, u_{i2}u_{j1} \in E_S | v_iv_j \in E\};$$
$$E_{S,2} = \{u_{i3}u_{j2} \in E_S | v_iv_j \in E \wedge v_i \in S\};$$
$$E_{S,3} = \{u_{j1}u_{i3} \in E_S | v_iv_j \in E \wedge v_i \in \overline{S}\};$$
$$E_{S,4} = \{u_{i1}u_{02}, u_{i2}u_{01} \in E_S | v_i \in V\};$$
$$E_{S,5} = \{u_{i3}u_{01} \in E_S | v_i \in V \wedge v_i \in S\};$$
$$E_{S,6} = \{u_{01}u_{i3}, \in E_S | v_i \in V \wedge v_j \in \overline{S}\};$$
$$E_{S,7} = \{u_{i3}u_{i3}, \in E_S | v_iv_j \in E \wedge v_i \in S \wedge v_j \in \overline{S}\};$$

Since the partition of all u_{i1}'s and u_{i2}'s is unrelated to the cut $(S,\overline{S})$, we get $|E_{S,1}| = 2|E|$ and $|E_{S,4}| = 2|V|$. Besides, for any u_{j1}, u_{j2} and u_{i3}, where $v_iv_j \in E$, no matter what the partition of u_{i3} belongs to, links $u_{j1}u_{i3}$ and $u_{j2}u_{i3}$ exist. Thus, $|E_{S,2}| + |E_{S,3}| = 2|E|$. Similarly, $|E_{S,5}| + |E_{S,6}| = |V|$. Moreover, for any pair of u_{i3} and $u_{j3}, u_{i3}u_{j3} \in E_{S,7}$ if and only if $v_iv_j \in (S,\overline{S})$. As a sequel, if $|(S,\overline{S})| \geqslant p$, we get $|E_S| \geqslant q = 4|E| + 3|V| + p$.

Conversely, for any $\rho = (V_F, V_S)$ of G_C, if it is feasible, there must be a pair of interfaces with different roles for each u_i. In addition, since G_C is identical, the order within a node can be arbitrary. Thus, we can restrict our concern to the case, where $u_{i1} \in V_S$ and $u_{i2} \in V_F \forall v_i \in V$. In this case, from the above observation, there must be $4|E| + 3|V|$ links between u_{j1}'s and u_{j2}'s, u_{j1}'s and u_{i3}'s, and u_{j2}'s and u_{i3}'s. So, if $|E_S| \geqslant q = 4|E| + 3|V| + p$, we can obtain a cut $(S,\overline{S})$ such that $|(S,\overline{S})|$ by choosing $v_i \in S, \forall u_{i3} \in V_S$ and $v_i \in \overline{S}$, $\forall u_{i3} \in V_F$.

Clearly, the constructions can be carried out in polynomial time, and $SL(q)$ is non-deterministic polynomial. Thus the $\mathcal{NP}$-completeness was established. □

Proof of Theorem 2. Firstly, we show that any resulting ρ is feasible. Without loss of generality, we assume that there is some feasible role assignment in the given G_C. For any two mesh nodes u_i and u_j, connected by some link $u_{ir}u_{jt}$ in G_C, because G_C is identical, there must be a link between any pair of interfaces on u_i and u_j in G_C. Besides, with Step 2, at least one pair of fixed and switchable interfaces has been assigned to both u_i and u_j. Hence, the two nodes can transmit to and receive from each other in the resulting $\overline{G}_S$.

On the other hand, for any partial solution (V_F, V_S), we denote *TNL* as the total number of links in G_C which are non-switchable in G_S, i.e.

$$TNL = \left(\sum_{u_{ir} \in V_F} N_{ir}(G_C) \cap V_F + \sum_{u_{ir} \in V_S} N_{ir}(G_C) \cap V_S \right) \Big/ 2.$$

In Step 2 and Step 3, if an interface v_{ir} is assigned to V_F, it means that $|N_{ir}(G_C) \cap V_F| < |N_{ir}(G_C) \cap V_S|$ and the *TSL* is increased by $|N_{ir}(G_C) \cap V_S|$; otherwise, it must be that $|N_{ir}(G_C) \cap V_F| \geqslant |N_{ir}(G_C) \cap V_S|$ and the increment to the *TSL* is $|N_{ir}(G_C) \cap V_S|$. In other words, no matter v_{ir} is assigned to V_F or V_S, the *TNL* and *TSL* are increased, respectively by $\min\{|N_{ir}(G_C) \cap V_S|, |N_{ir}(G_C) \cap V_F|\}$ and $\max\{|N_{ir}(G_C) \cap V_S|, |N_{ir}(G_C) \cap V_F|\}$. Consequently, at termination, we get $TNL < TSL$. Let TSL^* be the optimal. It is obvious that $TSL^* \geqslant TNL + TSL = |E_C|$. In addition, due to the fact that $TNL < TSL$, the worst case occurs when TNL approaches TSL. Hence, we get $TSL^* \geqslant TNL + TSL < 2TSL$, i.e. $TSL/TSL^* > 1/2$. □

Proof of Theorem 3. For any K'-color assignment σ' of G', if we assign the channels on V_{F} such that

$$\chi(u_{ir}) = \sigma'(v_{ir}), \quad \forall u_{ir} \in V_{\mathrm{F}}, \tag{8}$$

it clearly satisfies that

$$\sum_{(u_{ir},u_{jt})\in V_{\mathrm{F}}\times V_{\mathrm{F}}:\chi(u_{ir})=\chi(u_{jt})} \tau(u_{ir},u_{jt}) = \sum_{v_{ir}v_{jt}\in E':\sigma'(v_{ir})=\sigma'(v_{jt})} w'(v_{ir},v_{jt}). \tag{9}$$

Combining Eq. (4) with Eq. (9), we have

$$TIL = \sum_{v_{ir}v_{jt}\in E':\sigma(v_{ir})=\sigma(v_{jt})} w(v_{ir},v_{jt}).$$

On the contrary, for any channel assignment χ of V_{F}, if we set $\sigma(v_{ir}) = \chi(u_{ir}), \forall v_{ir} \in V'$, Eq. (9) still holds, which is also equal to the W in Eq. (2). Hence, the statement is proved. □

References

[1] I.F. Akyildiz, X. Wang, Weilin, Wireless mesh networks: a survey, Computer Network and ISDN System 47 (2005) 445–487.
[2] M. Kodialam, T. Nandagopal, Characterizing achievable rates in multi-hop wireless mesh networks with orthogonal channels, IEEE/ACM Transaction on Networking 13 (4) (2005) 868–880.
[3] S.L. Wu, C.Y. Lin, Y.C. Tseng, Jang-Ping Sheu, A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks, in: Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks, 2000, pp. 232–237.
[4] P. Kyasanur, N.H. Vaidya, Capacity of multi-channel wireless networks: impact of number of channels and interfaces, in: Proceedings of ACM MobiCom'05, 2005, pp. 43–57.
[5] 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.
[6] J. Crichigno, M.Y. Wu, W. Shu, Protocol and architecture for channel assignment in wireless mesh networks, Ad Hoc Networks 6 (2008) 1051–1077.
[7] H. Skalli, S. Ghosh, S.K. Das, L. Lenzini, M. Conti, Channel assignment strategies for multiradio wireless mesh networks issues and solutions, IEEE Communications Magazine 45 (11) (2007) 86–95.
[8] P. Kyasanur, N.H. Vaidya, Routing and interface assignment in multi-channel multi-interface wireless networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference, vol. 4, 2005, pp. 2051–2056.
[9] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, 2004, pp. 114–128.
[10] A.K. Das, H.M.K Alazemi, R. Vijayakumar, S. Roy, Optimization models for fixed channel assignment in wireless mesh networks with multiple radios, in: Proceedings of IEEE SECON, 2005, pp. 463–474.
[11] M. Alicherry, R. Bhatia, L. Li, Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks, in: Proceedings of the ACM MobiCom, 2005, pp. 58–72.
[12] N. Jain, S.R. Das, A. Nasipuri, A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless network, in: Proceedings of 10th International Conference on Computer Communications and Networks (iCCCN'01), 2001, pp. 432–439.
[13] J. So, N. Vaidy, Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver, in: Proceedings of MobiHoc'04, 2004, pp. 222–233.
[14] P. Bahl, R. Chandra, J. Dunagan, SSCH: Slotted seed channel hopping for capacity improvement in IEEE 802.11 Ad-Hoc Networks, in: Proceedings of ACM MobiCom, 2004, pp. 216–230.
[15] A. Raniwala, T.C. Chiueh, Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network, in: Proceedings of IEEE INFOCOM, 2005, pp. 13–17.
[16] P. Kyasanur, N.H. Vaidya, Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks, ACM SIG-MOBILE Mobile Computing and Communications Review 10 (2006) 31–43.
[17] P. Kyasanur, J. So, C. Chereddi, N.H. Vaidya, Multichannel mesh networks: challenges and protocols, IEEE Wireless Communications 13 (2) (2006) 30–36.
[18] C. Chereddi, P. Kyasanur, N.H. Vaidya, Design and implementation of a multi-channel multi-interface network, in: Proceedings of ACM REALMAN, 2006, pp. 23–30.
[19] C.Y. Li, A.A. K Jeng, R.H. Jan, A MAC protocol for multi-channel multi-interface wireless mesh network using hybrid channel assignment scheme, Journal of Information Science and Engineering 23 (4) (2007) 1041–1055.
[20] R.M. Karp, Reducibility among combinatorial problems, Complexity of Computer Computations, Plenum Press, New York, pp. 85–103.
[21] S.K. Sahni, T.F. Gonzalez, P-complete approximation problems, Journal of ACM 23 (3) (1976) 555–565.
[22] N. Garg, V.V. Vazirani, M. Yannakakis, Approximate max-flow min-(multi) cut theorems and their applications, SIAM Journal of Computing 25 (1996) 235–251.
[23] V. Kann, S. Khanna, J. Lagergren, A. Panconesi, On the hardness of approximating MAX k-CUT and its dual, Chicago Journal of Theoretical Computer Science 2 (1997) 1–18.
[24] S. Avallone, I.F. Akyildiz, G. Ventre, A channel and rate assignment algorithm and a Layer-2.5 forwarding paradigm for multi-radio wireless mesh networks, IEEE/ACM Transactions on Networking 17 (1) (2009) 267–280.
[25] A. Adya, P. Bahl, J. Padhye, A. Wolman, L. Zhou, A multi radio unification protocol for IEEE 802.11 wireless networks, in: Proceedings of BROADNETS, 2004, pp. 344–354.
[26] M.K. Marina, S.R. Das, A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks, in: Proceedings of BROADNETS, 2005, pp. 381–390.

Andy An-Kai Jeng received the BS degree in statistics from Tamkang University, Taiwan, in 2001, the MS degree in management information systems from National Chi Nan University, Taiwan,in 2003, and Ph.D. degree in the computer science from National Chiao Tung University, Taiwan, in 2007, where is currently a post-doctoral researcher. His research interests include wireless networks, distributed algorithm design and analysis, scheduling theory, and operations research.

Rong-Hong Jan received the B.S. and M.S. degrees in Industrial Engineering, and the Ph.D. degree in Computer Science from National Tsing Hua University, Taiwan, in 1979, 1983, and 1987, respectively. He joined the Department of Computer and Information Science, National Chiao Tung University, in 1987, where he is currently a Prof. During 1991–1992, he was a Visiting Associate Professor in the Department of Computer Science, University of Maryland, College Park, MD. His research interests include wireless networks, mobile computing, distributed systems, network reliability, and operations research.