Ad Hoc Networks 10 (2012) 401–420

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

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

On a joint temporal–spatial multi-channel assignment and routing scheme in resource-constrained wireless mesh networks

Yan Jin [a,*], Weiping Wang [a], Yingtao Jiang [b], Mei Yang [b]

[a] Institute of Computer Technology, Chinese Academy of Sciences, China
[b] Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, USA

ARTICLE INFO

Article history:
Received 24 January 2011
Received in revised form 1 August 2011
Accepted 3 August 2011
Available online 17 August 2011

Keywords:
Wireless mesh networks (WMNs)
Channel assignment
Temporal
Spatial
Routing
Linear programming (LP)
Throughput

ABSTRACT

Use of multiple orthogonal channels can significantly improve network throughput of multi-hop wireless mesh networks (WMNs). In these WMNs where multiple channels are available, channel assignment is done either in a centralized manner, which unfortunately shows a poor scalability with respect to the increase of network size, or in a distributed manner, where at least one channel has to be dedicated for exchanging necessary control messages or time synchronization has to be utilized for managing the duration of data packet transmission, causing excessive system overhead and waste of bandwidth resource. In this paper, we first formulate multi-channel assignment as a NP-hard optimization problem. Then a distributed, heuristic temporal–spatial multi-channel assignment and routing scheme is proposed, assuming every wireless node in the network is equipped with a single-radio interface. Here the gateway node is set to use all the channels sequentially in a round-robin fashion. This temporal scheme ensures all the nodes that need to directly communicate with the gateway node shall have a fair access to it. For those non-gateway nodes, a spatial scheme where channels are assigned based on their neighbors' channel usage is adopted to exploit parallel communications and avoid channel interference among nodes. Furthermore, since the routing factors, including channel usage of neighbor nodes, node hop count, node memory size, and node communication history, are all considered along with the channel assignment, network performance, measured by packet delivery latency, channel usage ratio, and memory usage ratio, tends to be considerably enhanced. The simulation results have confirmed that, compared with a couple of well-known multi-channel assignment schemes, such as LCM [21] and ROMA [15], the proposed scheme shows substantial improvement in network throughput with a very modest collision level. In addition, the proposed scheme is highly scalable as the algorithm complexity is only linearly dependent on the total number of channels that are available in the network and the number of neighbors that a network node directly connects to.

1. Introduction

Wireless mesh networks (WMNs) have emerged to meet the ever increasing demand for better and seamless network services [1]. In a WMN, due to limited radio transmission range, each wireless node has to utilize other nodes to communicate with gateway nodes in distance through a multi-hop communication path. However, when the number of hops in a communication path increases, the network performance tends to degrade sharply as a result of the increased channel contention/collision rate and the time experienced by a packet in the network [2]. To address this performance problem, many approaches targeting various protocol layers, spanning the application

* Corresponding author.
E-mail address: jinyanhit@gmail.com (Y. Jin).

1570-8705/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2011.08.001

layer [3], the network layer [4], the MAC layer [5], and the physical layer [6], have been proposed. Across these multiple layers, the MAC layer design is yet the most critical one as it directly builds upon the behavior of wireless communication [7]. Currently, the poor network throughput, introduced by the limited channel bandwidth, becomes one of the biggest performance issues related to the MAC layer protocols. Fortunately, up to 12 orthogonal (non-overlapping) channels in IEEE 802.11a are available among neighboring nodes [8], which provides the possibility of alleviating the network throughput degradation problem. Nevertheless, substantial improvement of network throughput cannot be readily assumed unless these channels are properly assigned.

Various schemes that attempt to make efficient use of multiple orthogonal channels in WMNs have been reported in the literature [9–24]. These schemes can be broadly categorized into two types: the centralized schemes [9–13,18] and the distributed schemes [14–16,19–23]. For centralized channel assignment schemes, channel assignment is performed at one central location and global information needs to be known before assigning channels. As opposed to the centralized schemes, channel assignment followed by different distributed schemes is done at each node independently and only local information is needed for making on-the-fly channel assignment decisions.

Assuming that each node is equipped with multiple radio interfaces, Huang et al. investigated the capability that WMNs can offer the wireless nodes to transmit/receive packets across the gateways over multiple orthogonal channels [24]. The centralized channel assignment schemes suggested in [9] and [10] adopt linear programming (LP) and graph theory to assign a channel to each communication node's interface. In recent years, realizing the cross-dependency between channel assignment and routing, a few centralized, joint multi-channel assignment and routing schemes have been proposed in [11–13]. However, these centralized solutions are so rigid that they cannot accommodate the changing traffic load. In contrast, a distributed, hybrid multi-channel protocol (HMCP) was introduced, where the interfaces of each network node are divided into fixed ones and switchable ones [14]. As most of the traffic in a typical WMN is directed from/to the wired network via gateway(s), each non-gateway node needs to discover a path to the nearest gateway node. Herein, Dhananjay et al. proposed ROMA, a distributed multi-channel assignment and routing algorithm in dual-radio tree-based WMNs [15]. In ROMA, as each node needs to assign the channels to its radios based on the associated gateway's channel sequence, time synchronization is required. In addition, to find a routing path that has the combination of lower path overhead and higher path performance toward the gateway, each node uses link variation metric and external load metric. Another tree-based distributed scheme, called rate-based channel assignment (RB-CA) protocol, is proposed by Kim and Suh [16]. Instead of enhancing network performance, the goal of [16] is to alleviate performance anomaly by using multiple channels in WMNs.

As a matter of fact, having multiple radio interfaces can be too expensive for small and low-cost devices. In addition, a node's transmitting radio might cause unpredictable interfere with its receiving radio, unless these two radios are separated by a sufficient distance [15]. As a result, nodes are typically equipped with just one single-radio interface in resource-constrained (i.e., low cost and/or low power) WMNs. Even with a single radio interface per node, it has been found that up to $O(\log n)$ channels still can be fully utilized with n nodes [17]. This has motivated a number of research works to develop efficient channel assignment schemes for this type of networks.

Using graph edge coloring approach, Aryafar et al. introduced a centralized Distance-1 Constrained Channel Assignment (D1C-CA) algorithm and the corresponding heuristic solution [18]. Bahl et al. proposed a distributed single-radio protocol called Slotted Seeded Channel Hopping (SSCH) [19] that each node switches on its radio interface by a pseudo-random sequence at each fixed time slot. In this protocol, any two neighbor nodes will have their channels overlapped periodically, and the two nodes can communicate with each other when their channels overlap. SSCH, as it requires strict time synchronization to implement time slotting, is infeasible for large networks. So and Vaidya thus proposed another distributed, time-synchronized multi-channel assignment protocol MMAC [20]. The beacon interval in this MMAC protocol is divided into multiple cycles, and each cycle is composed of a fixed control phase and a fixed data exchange phase. During the control phase, all nodes switch to a dedicated channel and negotiate an appropriate channel to be used in the data exchange phase. The data exchange phase is followed to let the data packets be transmitted over the negotiated channel. Similar to the protocol proposed in [19], MMAC protocol requires global time synchronization for every cycle so that all the nodes can begin their beacon intervals at the same time instance, which suffers from increased system overhead and is hard to achieve optimized use of channel bandwidth. As an improvement to MMAC, Maheshwari et al. proposed a distributed, time synchronization-free protocol called Local Coordinated-based Multichannel (LCM) [21], where the intervals of the control phase and the data exchange phase are adjustable according to the local network load. From the view of alleviating flow starvation, Shi et al. devised a distributed, Asynchronous Multi-channel single-radio Coordination Protocol (AMCP) [22]. However, both LCM and AMCP still require a dedicated channel for the sole purpose of channel coordination, which inevitably causes the waste of valuable network bandwidth, and this problem can be more pronounced when the number of available channels is limited. For example, as there are only three available orthogonal channels in IEEE 802.11b, having one dedicated channel results in 33% of the total bandwidth spent on the control overhead. In this case, as only two available data channels are used for simultaneous data communications, the number of control messages generated tends to be really small, which causes excess idle time slots in the control channel. On the other side, when there are a large number of data channels, having just one dedicated channel may not provide enough bandwidth to exchange many control messages thus generated. In this case, nodes have to hold their data transmissions over data channels after their control messages can finally get through the control channel, which obviously leads to poor utilization of data channels. An intuitive addition of more dedicated channels will not help improve the network performance, as more multi-channel hidden-terminal problems tend to be created [20]. In a simple word, using dedicated channel(s) is not an efficient solution to assign multiple channels.

In this paper, we first mathematically formulate the multi-channel assignment in WMNs as an optimization problem. To overcome the deficiencies in aforementioned known schemes, we propose a distributed, heuristic temporal–spatial channel assignment and routing scheme that can achieve higher network throughput. The proposed scheme has the following distinct advantages.

(i) Control messages and data packets can be separately transmitted over different channels, which help increase the channel bandwidth utilization.
(ii) Multiple orthogonal channels are assigned to the communication nodes, not to the packet routes/flows. For example, a packet received by an intermediate node u through channel i may be forwarded by node u to its next-hop neighbor through a different channel, say channel j. That is, each communication node only needs to find the next hop node for each packet to be transmitted, which helps decrease the complexity of the algorithms running on these resource-constrained nodes.
(iii) To provide each individual node a fair access to the gateway node, the gateway node utilizes all the available channels in a round-robin fashion to collect packets from its neighbors. For all other nodes, they simultaneously utilize different channels within their neighbors for communications.
(iv) Selection of nodes along a routing path considers multiple factors to boost the overall network performance. These factors include channel usage of node's neighbor nodes, node's memory size, node's hop count, and node's transmission history.

An initial version of above results can be found in [23].

The rest of the paper is organized as follows. Problem formulation and the solutions are provided in Sections 2 and 3, respectively. In Section 4, performance results are presented to compare the proposed channel assignment scheme with a couple of popular channel assignment schemes. Finally, the conclusions are summarized in Section 5.

2. Preliminaries and problem formulation

2.1. Notations and assumptions

Table 1 lists the notations used in this paper.

Assumptions:

(1) There are $|S_C|$ orthogonal channels, denoted as channels 1, 2, ..., and $|S_C|$. Each of these channels has a channel capacity of C. Since none of the channels is assumed to overlap with each other, the packets transmitted on one channel will not interfere any of other channels.
(2) Each node, say node u, is equipped with a single half-duplex transceiver whose radius is the same as its radio transmission range r_t. Each node stays on either busy state (i.e., the node is communicating with another node on its assigned channel $C_A(u)$) or idle state (i.e., the node is listening to its assigned channel $C_A(u)$) at any time instance. As the time needed for switching from one channel to another is finite but typically short (80 μs as reported in [19]), we assume that the channel switching can take place instantaneously.
(3) All the network nodes are randomly distributed. Each non-gateway node has a limited memory size and may function as a source node which generates data packets. Initially, we have $M_a(u) = M(u)$ for any non-gateway node u. The memory size of the gateway node is assumed to be unlimited.
(4) The transmitter–receiver conflict avoidance (TRCA) model [25] is adopted. In this model, the transmission on channel i over a link is successful when all the neighbors of the sender and the receiver are silent on channel i for the entire transmission duration.
(5) All the nodes maintain their own clocks independently. Each node's internal clock is accurate but it may not be synchronized precisely to the standard time.

Table 1
Notations.

Symbol	Description
$\lvert \cdot \rvert$	$\lvert \cdot \rvert$ represents the cardinality of a set
V	Set of all wireless nodes in the network
E	Set of all wireless links in the network
$G(N,E)$	Undirected topology graph that is composed of node set V and edge set E
S_C	Set of non-overlapping channels, i.e., $S_C = \{1, 2, 3, \ldots, \lvert S_C \rvert\}$
C	Capacity of a channel
r_t	Radio transmission range of a node
$s(e)$, $r(e)$	Sender and receiver of a wireless link e ($e \in E$)
$C_A(u)$	Current assigned channel of node u
$M(u)$, $M_a(u)$	Memory size, currently available memory size of node u
$H(u)$	Hop count of node u
$H_{\max}$	Maximal hop count of all wireless nodes in graph G, i.e., $H_{\max} = max\{H(u), u \in V\}$
$S_N(u)$	Neighbor set of node u
Δ	Maximal degree of all wireless nodes in graph G, i.e., $\Delta = max\{\lvert S_N(u) \rvert, u \in V\}$

2.2. Network architecture

In this paper, we focus on the network architecture formed by climate change infrastructure. In this infrastructure, the nodes that collect the environmental data need to generate the traffic flows toward the gateway in timely fashion, so that the gateway can quickly analyze the received data and thus predict the climate change trend. As a result, these uplink traffic flows form a reverse spanning tree. In this application, it is not necessary to consider the downlink flows that go from the wired Internet to any wireless node. In addition, for the sake of presentation, we also limit the number of gateway nodes to just one.

Definition 1. A network that consists of a gateway node *GW* and a number of wireless nodes can be modeled by an undirected communication graph $G(V, E)$. Given a node $u \in V$ and a node $v \in V$, we have $(u, v) \in E$, if and only if $\text{dis}(u, v) \leqslant r_t$, where $\text{dis}(u, v)$ is the Euclidean distance between node u and node v. That is, to establish a direct communication between any two nodes, the distance between them has to be within their *radio ranges* r_t, and they also need to have a common channel assigned to their interfaces. In this paper, G is assumed to be connected.

Definition 2. The hop count of node u is h, provided the least number of hops traversed by a packet between node u and the gateway node *GW* in G (defined in Definition 1) is h. Note that the hop count of node *GW* is 0.

Definition 3. A node is called a *non-gateway node* if it is not a gateway node.

Definition 4. A node is called a *1-hop sender node* if it is a sender node and its hop count is one.

Definition 5. A node is called a *non 1-hop sender node* if it is a sender node and its hop count is larger than one.

Definition 6. A node is called a *non-gateway, non-sender node* if it is a non-gateway node and it is not a sender node.

Each node can find all its neighbors by using a distributed algorithm shown in Fig. 1. Initially, all nodes need to switch to a pre-determined, common channel (e.g., channel 1) to listen to HELLO control messages (line 1). A HELLO message sent from a neighbor of node u includes the neighbor's hop count information. By setting up a back-off system timer T_S (line 14), node u will have sufficient amount of time to receive the HELLO messages coming from all of its neighbors. That is, node u shall be provided with all its neighbors' hop count information so that it can eventually determine its hop count to the gateway node. Although a large back-off timer may increase the total time required for network formation, it is an acceptable solution as the nodes in WMNs are stationary and this algorithm runs only once at the initial topology establishment stage. On the other hand, since node u will not broadcast its HELLO message until this back-off timer expires, this timer mechanism helps avoid generating excess HELLO messages.

After executing the algorithm listed in Fig. 1, any node in the network, say node u, shall obtain its hop count $H(u)$, maintain a neighbor set $S_N(u)$ in its local memory, and randomly select a channel from S_C as its current assigned channel. Additionally, by exchanging the hop count information with its neighbor nodes, any node can get the maximal hop count of all wireless nodes, $H_{\max}$.

2.3. Problem formulation

The multi-channel assignment optimization problem (MCAO): Given a topology graph $G(V,E)$, channel capacity C associated with each wireless link e ($e \in E$), and a source node set S with a desired flow rate $fr(k)$ originated from each source node k ($k \in S$), assign the channels to the nodes such that the gateway node can receive the data packets generated by all the source nodes with the minimal interval.

Before achieving this problem, we first divide the entire data communication procedure into time slots and intro-

Neighbor discovery algorithm (*u*: node)

1. Switch to a pre-determined, common channel to listen HELLO messages;
2. $S_N(u) \leftarrow \Phi$, bInitialized $\leftarrow$ 0; // Φ denotes a empty set
3. **if** u is a non-gateway node **then**
4. $H(u) \leftarrow \infty$;
5. **elseif** u is a gateway node **then** { // hop count of gateway node is 0
6. $H(u) \leftarrow 0$;
7.   Broadcast a HELLO message, piggybacked with value of $H(u)$;
8.   **exit**; }
9. **while** ((bInitialized = 1 **and** GetCurrentTime() < T_S) **or** $S_N(u)$ is Φ) {
10.   **if** (node u receives a HELLO message from node v) **then** {
11. $S_N(u) \leftarrow S_N(u) \cup \{v\}$;
12.   **if** $|S_N(u)| = 1$ **then** {
13.   bInitialized $\leftarrow$ 1;
14.   Initialize a back-off system timer T_S; } } }
15. $H(u) \leftarrow min\ \{\ min\ \{\ H(w) |\ w \in S_N(u)\ \} + 1, H(u)\ \}$;
16. Broadcast a HELLO message, piggybacked with value of $H(u)$;
17. $C_A(u) \leftarrow rand(S_C)$; // randomly select a member from S_C as node u's assigned channel

Fig. 1. Neighbor discovery algorithm.

duce two variables: $f^t(u, v)$, which is defined as the flow rate from node u to its neighbor node v at time slot t, and a 0–1 indicator variable, $x_i^t(e)$, which is defined as

$$x_i^t(e) = \begin{cases} 1, & \text{if link } e \text{ is active on channel } i \text{ in time slot } t \\ 0, & \text{otherwise} \end{cases}.$$

The MCAO problem can be formulated as follows:
Minimize T
Subject to

$$\sum_t \sum_{e=(u,v)\in E, v=r(e)} f^t(u,v) = \sum_t \sum_{e=(u,v)\in E, v=s(e)} f^t(v,u), \quad \forall u \in V - S - \{GW\}, \quad \forall t \leqslant T \tag{1}$$

$$\sum_t \sum_{e=(k,v)\in E, v=r(e)} f^t(k,v) = fr(k), \qquad \forall k \in S, \quad \forall t \leqslant T \tag{2}$$

$$\sum_t \sum_{e=(v,GW)\in E, GW=r(e)} f^t(v,GW) = \sum_{k\in S} fr(k), \quad \forall t \leqslant T \tag{3}$$

$$f^t(u,v) \leqslant C, \qquad \forall e=(u,v) \in E, \quad \forall t \leqslant T \tag{4}$$

$$\sum_{e=(u,v)\in E} \sum_{i\in S_C} x_i^t(e) \leqslant 1, \quad \forall u \in V, \quad \forall t \leqslant T \tag{5}$$

$$\sum_{e=(u,v)\cup(p,q)\in E} x_i^t(e) \leqslant 1, \qquad \forall u \in V, \quad \forall p \in S_N(u), \quad \forall i \in S_C, \quad \forall t \leqslant T \tag{6}$$

Here we use the interval T to measure the network performance and attempt to optimize it. Constraint (1) sets the *flow conservation constraint* for any node u which is not the source node or the gateway node. Constraint (2) sets the *flow conservation constraint* for any node k which is a source node, and constraint (3) sets the *flow conservation constraint* for gateway node GW. Constraint (4) ensures no link capacity is violated. Constraint (5) sets the *node radio constraint*. Constraint (6) sets the *interference constraint*. As this problem is inherently NP-hard [12], we propose a distributed, heuristic multi-channel assignment and routing algorithm to solve this problem in the next section.

3. Distributed temporal–spatial multi-channel assignment and routing scheme

In WMNs, each non-gateway node functions either as a sender or as a receiver. In conventional single-channel networks, as shown in Fig. 2a, when node A is communicating with node B on channel 1, the only channel available in the network, for all of node A's neighbors except B, (nodes C, D, E and F) even if they have data packets to transmit, they have to hold their transmissions to avoid possible channel collisions. In a sharp contrast, when there are multiple channels in the network, as shown in Fig. 2b, several parallel communications are possible, like node C can communicate with node D on channel 2 and node F can communicate with node E on channel 3, which obviously can help improve the network throughput. In this case, a

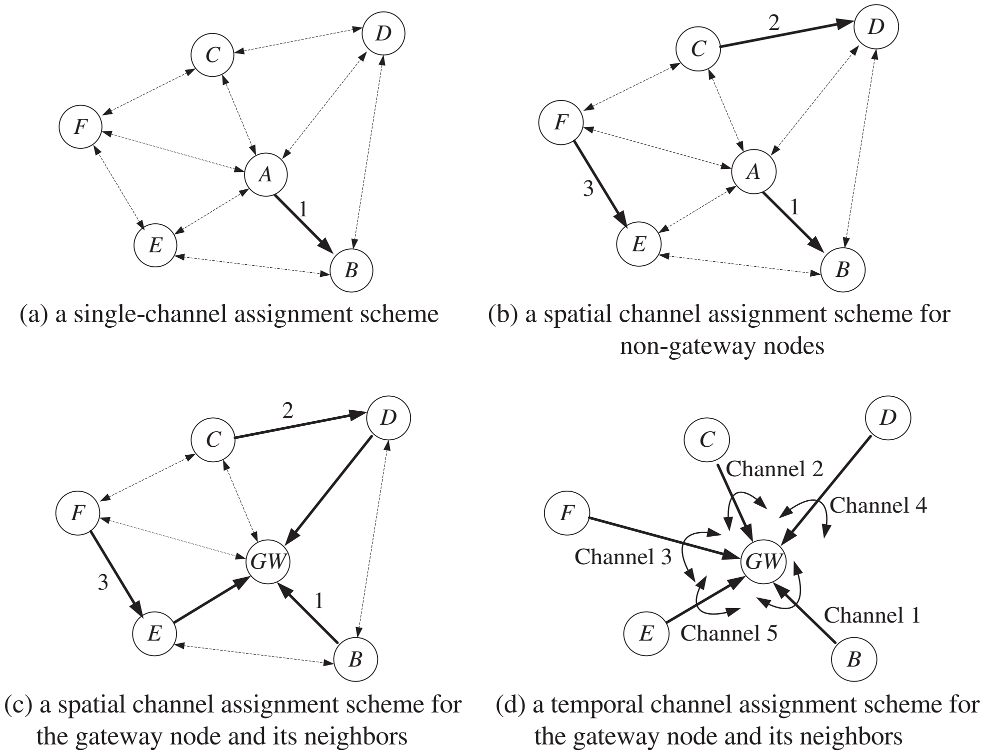


Fig. 2. Channel assignment scheme for different nodes.

collision domain in a single-channel network is logically replaced by multiple collision domains in a multi-channel network with each channel operating in a different frequency band.

In a WMN, a typical gateway node sinks all the data packets originated from other nodes. Consider the case that the gateway node *GW* is receiving packets from one of its neighbors (Fig. 2c), say node *B*. If the neighbors of *GW* follow the same spatial channel assignment scheme that was applied to Fig. 2b, one can see that the transmitted packets that are coming from 1-hop nodes (nodes *C* and *F*) have to traverse at least two hops to reach *GW* ($C \to D \to GW$ and $F \to E \to GW$ in Fig. 2c), resulting in significant bandwidth waste and degradation of gateway throughput in terms of the total bits received by the gateway node per second. Alternatively, if the gateway node uses all the channels in a round robin fashion as shown in Fig. 2d, then the transmitted packets that are coming from all the 1-hop nodes only need to traverse one hop to reach *GW*, which indicates that the gateway throughput can be significantly improved. More importantly, compared with non-gateway node, as the gateway node now no longer needs to maintain the channel usage information of its neighbor nodes in its local memory, storage space of the gateway node can be saved significantly. In addition, a temporal channel assignment scheme followed in Fig. 2d can assure a fair access of all the available channels for the gateway node.

The above observations have motivated us to adopt different multi-channel assignment schemes for non-gateway nodes and the gateway node. Moreover, as there is a strong dependency between radio channel assignment and packet routing, the proposed channel assignment scheme is considered along with routing, which involves various routing factors such as neighbors' channel usage, node memory size, node hop count, and node communication history. Details of the proposed algorithms are provided in the following.

3.1. Non-gateway node: a spatial scheme

3.1.1. Algorithm description

Each non-gateway node needs to maintain a channel usage table (CUT) to record its neighbors' channel usage information. Initially, there is no traffic in the network, and thus every node's CUT is empty. When a node detects that a neighbor is transmitting or receiving, a new entry, which includes four fields: (i) neighbor ID (NID), (ii) neighbor working channel (NWC), (iii) communication beginning time (CBT), and (iv) announced duration period (ADP) on this working channel, will be created and inserted into this node's CUT. When a node has data to transmit, it will check its CUT to find an available channel before transmission. At time instance *t*, node *u*'s CUT may contain information similar to what is given as Table 2. In this example, node *u* holds the following information: node v_i initiated the communication on channel *i* at t_i, and the duration is adp_i seconds ($i = 1, 2$). Hence, to avoid channel collision, node *u* has to use another channel if it has an immediate need to send out data packets. After time instance $t_i + adp_i$, node *u* can remove the entry of v_i from its CUT and channel *i* becomes available again.

Table 2
CUT of node *u* at *t*.

NID	NWC	CBT	ADP
v_1	1	t_1	adp_1
v_2	2	t_2	adp_2

A non-gateway node can be functionally categorized into two types: sender nodes or non-sender nodes. Different strategies, with their control messages and timers summarized in Table 3, are adopted to accommodate these two types of nodes.

The algorithm periodically performed at a non 1-hop sender node, say node *u*, is presented in Fig. 3, which involves the following three major steps.

Step 1: Based on the content stored in CUT, each non 1-hop sender node *u* randomly selects a channel from set $S_{C\_u} \setminus S_{N\_WC}(u)$ as its current assigned channel.

Step 2: If there is no available channel or idle neighbor at the current time instance, node *u* will have to defer its transmission until one of its neighbors releases a channel (lines 5–8 in Fig. 3); otherwise, node *u* can proceed to *Step 3*.

Step 3: In this case, node *u* will broadcast a RTT message on its current assigned channel $C_A(u)$ and initializes a system timer T_{neg}. The value of T_{neg} can be determined by adding the maximal back-off time to the time needed to transmit the control messages and the guard time between two adjacent data packets. There are two scenarios to consider:If node *u* receives a RTR message from one of its neighbors within T_{neg}, it means node *u* has found an available receiver on channel $C_A(u)$ and it should ensure that all the *u*'s neighbors are aware that channel $C_A(u)$ will be taken by *u*. Correspondingly, node *u* performs "active scanning" by broadcasting a control message CRE on all the channels (line 14 in Fig. 3). This way, the multi-channel hidden terminal problem as well as the deafness problem [5] can be avoided. After broadcasting a CRE message, node *u* can begin its data transmission.On the other hand, if node *u* does not receive a RTR message after T_{neg} expires, it will come to the conclusion that no connection with any neighbor on channel $C_A(u)$ can be established. In this case, node *u* will then have to go back to search for another channel (lines 18–20 in Fig. 3). If there is no available

Table 3
Messages and timers.

Message	Message description	Timer	Timer description
RTT	Request-To-Transmit message sent by a sender	T_{neg}	System timer triggered by a sender
RTR	Request-To-Reply message sent by a receiver	$T(u)$	Timer triggered by a potential receiver *u*
CRE	Channel-REserve message sent by a sender		

Spatial_scheme (*u: non 1-hop sender node*)

0. $S_{C\_u} \leftarrow S_C$; // S_{C_u} is a temporary variable that stores the available channel set of node u

Step 1:

1. $S_{N\_WC}(u) \leftarrow \bigcup_i u.\text{CUT}[i].\text{NWC}$; // $S_{N\_WC}(u)$ is the set of the working channels of node u's neighbors
2. $S_{N\_WI}(u) \leftarrow \bigcup_i u.\text{CUT}[i].\text{NID}$; // $S_{N\_WI}(u)$ is the set of the ids of node u's neighbors that are working
3. **if** $C_A(u) \in S_{N\_WC}(u)$ **then**
4. $\quad C_A(u) \leftarrow rand(S_{C\_u} \backslash S_{N\_WC}(u))$;

Step 2:

5. **if** ($S_{N\_WC}(u) = S_{C\_u}$ **or** $S_{N\_WI}(u) = S_N(u)$) **then** { // node u has no available channel or available neighbor
6. $\quad j \leftarrow \arg\min\{ u.\text{CUT}[i].\text{CBT} + u.\text{CUT}[i].\text{ADP}$;
7. $\quad$ Sleep ($u.\text{CUT}[j].\text{CBT} + u.\text{CUT}[j].\text{ADP} - t$);
8. $\quad$ **goto** line 1; } // node u goes back to execute *Step 1*

Step 3:

9. Broadcast a RTT message on channel $C_A(u)$, piggybacked with value of $M(u) - M_a(u)$;
10. Initialize a timer T_{neg};
11. **while** (current clock time $t_{\text{cur}} < T_{\text{neg}}$) {
12. $\quad$ **if** (node u receives a RTR message M_v from a node v) **then** {
13. $\quad\quad$ Cancel the timer T_{neg};
14. $\quad\quad$ Broadcast a CRE message on all the channels, piggybacked with value of $C_A(u)$, value of current clock time t as the CBT, and value of M_v.ADP as the ADP;
15. $\quad\quad$ Insert a 4-field record (v, $C_A(u)$, t, M_v.ADP) into node u's CUT;
16. $\quad\quad$ Transmit data packets to node v;
17. $\quad\quad$ **exit**; } }
18. $S_{C\_u} \leftarrow S_{C\_u} \backslash \{C_A(u)\}$;
19. **if** $S_{C\_u} \,!= \Phi$ then
20. $\quad$ **goto** line 1;
21. **else exit**;

Fig. 3. Spatial scheme for a non 1-hop sender node u.

channel to choose from, node u will have to halt this spatial scheme (line 21 in Fig. 3)

The algorithm performed at a non-gateway, non-sender node u is presented in Fig. 4, which involves the following three major steps.

Step 1: This step is the same as the one in non-gateway, non 1-hop sender case shown in Fig. 3.

Step 2: If node u receives a RTT message from one of its neighbors, say node m, node u will defer broadcasting its RTR message when a back-off timer $T(u)$ expires ($0 \leqslant T(u) \leqslant T_{\text{neg}}$). The value of $T(u)$ is calculated by:

$$T(u) = T_{\text{neg}}[\alpha(1 - M_a(u)/M(u)) + \beta(1 - |S_C \setminus S_{N\_WC}(u)|/|S_C|) + (1 - \alpha - \beta)H(u)/H_{\max}] \quad (7)$$

where α and β are two adjustable parameters ($\alpha \geqslant 0$, $\beta \geqslant 0$, $0 \leqslant \alpha + \beta \leqslant 1$), $|S_C \setminus S_{N\_WC}(u)|$ is the number of currently available channels for node u, and $H_{\max}$ is the maximal hop count of all wireless nodes. Based on Eq. (7), a node with a lower memory usage ratio, a larger number of available channels, and a smaller hop count has a higher probability to be selected as the receiver. Similar to *Step 3* in Fig. 3, the receiver needs to broadcast a control message RTR on all the channels to inform all its neighbors of this connection, as shown in line 19 of Fig. 4.

Step 3: For a node that is not a receiver, it will need to update its CUT once it receives a RTR or a CRE message from any of its neighbor node.

Note that in *Step 3* of Fig. 4, collisions may occur at a pair of nodes' neighbors, provided these neighbors are communicating with other nodes when they receive a CRE or a RTR message. That is, these nodes may not be able to update their CUTs timely regarding the channel usage information of their neighbors, which will cause a slight increase of collision ratio in the network. However, simulation results (in Section 5) have shown that the proposed scheme can deliver better network throughput even at the presence of a modest higher level of collisions.

3.1.2. An illustrative example

The algorithms shown in Figs. 3 and 4 are illustrated using a simple example that involves five non-gateway nodes (Fig. 5). Here any node, say u, is associated with (i) a set of channels occupied by node u's neighbors $S_{N\_WC}(u)$, (ii) node u's current assigned channel $C_A(u)$, (iii) node u's memory size $M(u)$, (iv) node u's current available memory size $M_a(u)$, and (v) node u's hop count $H(u)$ at time instance t. In the example shown in Fig. 5a, one can see that one of node B's neighbors, which is absent in this figure, is communicating with node B on channel 4.

Spatial_scheme (*u: non-gateway, non-sender node*)

Step 1:

1. $S_{N\_WC}(u) \leftarrow \bigcup_i u$.CUT[$i$].NWC; // $S_{N\_WC}(u)$ is the set of the working channels of node u's neighbors
2. $S_{N\_WI}(u) \leftarrow \bigcup_i u$.CUT[$i$].NID; // $S_{N\_WI}(u)$ is the set of the ids of node u's neighbors that are working
3. **if** $C_A(u) \in S_{N\_WC}(u)$ **then**
4. $C_A(u) \leftarrow rand(S_{C\_u} \backslash S_{N\_WC}(u))$;

Step 2:

5. **if** (an idle node u receives a RTT message from a node m **and** $H(u) \le H(m)$ **and** $M_a(u)$!= 0) **then** {
6. $T(u) \leftarrow T_{\text{neg}}[\,\alpha\,(1 - M_a(u)/M(u)) + \beta\,(1 - |S_C \backslash S_{N\_WC}(u)|/|S_C|) + (1 - \alpha - \beta)\,H(u)/H_{\max}]$;
7. Initialize a timer $T(u)$;
8. **while** (current clock timer $t_{\text{cur}} < T(u)$) {
9. **if** (node u receives a RTR message) **then** {
10. Cancel the timer $T(u)$;
11. **goto** line 20; }
12. **elseif** (node u receives a CRE message M_w from a node w) **then** {
13. Cancel the timer $T(u)$;
14. Insert a 4-field record (w, $C_A(u)$, M_w.CBT, M_w.ADP) into its CUT;
15. **goto** line 20; } }
16. Mark itself as a receiver node on channel $C_A(u)$;
17. Broadcast a RTR message, piggybacked with value of $min\{M(m) - M_a(m), M_a(u)\}/C$) as the ADP;
18. **if** (node u receives a CRE message M_p from a node p) {
19. Broadcast a RTR message on all the channels, piggybacked with value of channel $C_A(u)$, value of M_p.CBT as the CBT, and value of M_v.ADP as the ADP;
20. Insert a 4-field record (p, $C_A(u)$, M_p.CBT, M_p.ADP) into its CUT; } }

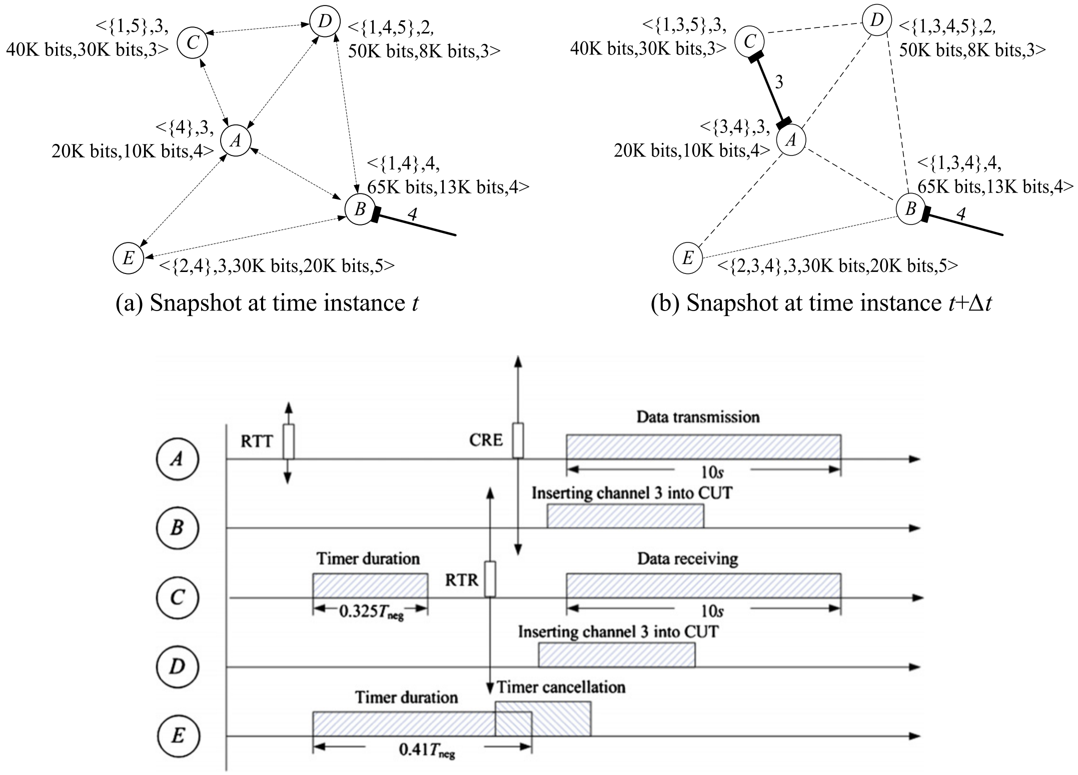
Step 3:

21. **if** (node u receives a RTR message or a CRE message M_q from a node q) **then**
22. Insert a 4-field record (q, $C(q)$, M_q.CBT, M_q.ADP) into node u's CUT;

Fig. 4. Spatial scheme for a non-gateway, non-sender node u.

Consider a case that a 4-hop sender node A plans to transmit its data packets. From the information given in Fig. 5a, node A knows that its assigned channel (i.e., $C_A(u) = 3$) is not being occupied by any neighbor. Then node A broadcasts a RTT message on channel 3, piggybacked with value of $M(A) - M_a(A)$ (=10 K bits), and initiates its timer. All the idle neighbors that are listening to channel 3 (node C and node E in this case) will receive this RTT message and initialize their own timers independently. Given $|S_C| = 5$, $C = 1$ K bps, $H_{\max} = 10$, $\alpha = 0.3$ and $\beta = 0.4$, based on Eq. (7), we have $T(C) = 0.325T_{\text{neg}}$ and $T(E) = 0.41T_{\text{neg}}$. Since $T(C) < T(E)$, node C will become a receiver by broadcasting a RTR message to establish a communication with node A on channel 3, according to line 17 of Fig. 4. The duration of this data communication session is $min\{M(A) - M_a(A), M_a(C)\}/C$ (=10 s). Once receiving the RTR message from node C, node A will try to reserve channel 3 by broadcasting a CRE message on all the channels (line 14 of Fig. 3). As stated in *Step 3* of Fig. 4, when node D receives a RTR message from node C or a CRE message from node A, node D will insert a record about channel 3's usage information into its CUT, which prohibits node D from using channel 3 until node C finishes its communication with node A. Correspondingly, all other non-sender nodes, nodes B and E in this case, that receive a CRE message or a RTR message will cancel their timers (if they have ever started) and then insert a record about channel 3 usage information into their respective CUTs. So after all these operations are completed by these five nodes, at time instance $t + \Delta t$, the status of these nodes has evolved to what is given in Fig. 5b. The complete time lines of above message sequence for these five nodes are shown in Fig. 5c.

Note that node D or E (shown in Fig. 5), which only received control messages but did not participate in data communication, actually could communicate with other nodes (not shown in Fig. 5) on another available channel when node A is communicating with node C on channel 3. That is, different data packets can be transmitted concurrently on different channels to exploit communication parallelism. After the transmission time of channel 3 expires, nodes B, D and E will remove the records of channel 3 from their respective CUTs. That is, any of these five nodes shall be able to utilize channel 3 afterwards without causing a channel collision.



(a) Snapshot at time instance t (b) Snapshot at time instance $t+\Delta t$

(c) Message sequence of the spatial scheme

Fig. 5. An example with five non-gateway nodes to illustrate the algorithms described in Figs. 3 and 4. Each node needs to know its neighbors' current working channels, determine a channel through which it can transmit, and store relevant information in its local memory. Taking node A in (a) as an example. Here {4} is the working channel set of node A's neighbor; channel 3 is the assigned channel to A; A has a total of 20 K bits of storage in its memory, of which 10 K bits is currently available; the hop count of node A is 4.

3.1.3. Analysis on the channel assignment delay

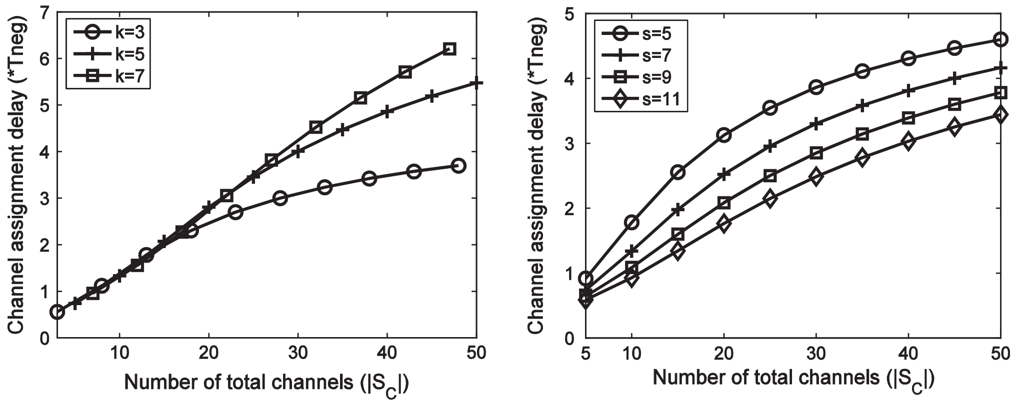
The proposed spatial scheme aims to effectively assign available channels to the nodes that need to communicate. Without loss of generality, we assume the number of a sender node u's idle neighbors (denoted as s, $s \leqslant |S_N(u)|$) and the number of node u's available channels (denoted as k, $k \leqslant |S_C|$) are fixed during a time interval. Then we will analyze the expected channel assignment delay that node u undertakes. Here the transmission duration of a control message can be ignored because the size of such a message tends to be too small.

Proposition 1. The expected channel assignment delay of any sender node u is: $T_{\text{delay}}(u) = T_{\text{neg}}\left[\frac{1}{2} + \sum_{i=1}^{k}\left(1-\frac{i}{|S_C|}\right)^s + \left(k-\frac{3}{2}\right)\left(1-\frac{k}{|S_C|}\right)^s\right]$, where s is the number of node u's idle neighbors and k is the number of node u's available channels.

Proof. Let H_i ($\overline{H_i}$) correspond to the event that none of node u's idle neighbors (at least one of node u's idle neighbors) stays on node u's assigned channel when node u executes *Step 3* of Fig. 3 at the ith iteration. Therefore, the expected channel assignment delay can be calculated as follows.

$$
\begin{aligned}
T_{delay}(u) &= \frac{1}{T_{\text{neg}}}\int_0^{T_{\text{neg}}} P_r(\overline{H}_1)tdt^{+}\sum_{i=1}^{k-1}\frac{1}{T_{\text{neg}}} \\
&\quad\times\int_0^{T_{\text{neg}}} P_r(H_1\cap H_2\cap\ldots\cap H_i\cap\overline{H_{i+1}})[iT_{\text{neg}}+t]dt \\
&= \frac{1}{T_{\text{neg}}}\int_0^{T_{\text{neg}}}\left[1-\left(1-\frac{1}{|S_C|}\right)^s\right]tdt^{+}\sum_{i=1}^{k-1}\frac{1}{T_{\text{neg}}} \\
&\quad\times\int_0^{T_{\text{neg}}}\left(1-\frac{1}{|S_C|}\right)^s\left(1-\frac{1}{|S_C|-1}\right)^s\ldots \\
&\quad\times\left(1-\frac{1}{|S_C|-(i-1)}\right)^s\left[1-\left(1-\frac{1}{|S_C|-i}\right)^s\right] \\
&\quad\times[iT_{\text{neg}}+t]dt = \frac{T_{\text{neg}}}{2}\left[1-\left(1-\frac{1}{|S_C|}\right)^s\right] \\
&\quad+\frac{T_{\text{neg}}}{2}\left[\left(1-\frac{1}{|S_C|}\right)^s-\left(1-\frac{k}{|S_C|}\right)^s\right] \\
&\quad+T_{\text{neg}}\left[\sum_{i=1}^{k-1}\left(1-\frac{1}{|S_C|}\right)^s+(k-1)\left(1-\frac{k}{|S_C|}\right)^s\right] \\
&= T_{\text{neg}}\left[\frac{1}{2}+\sum_{i=1}^{k-1}\left(1-\frac{1}{|S_C|}\right)^s+\left(k-\frac{1}{2}\right)\left(1-\frac{k}{|S_C|}\right)^s\right]
\end{aligned}
\tag{8}
$$

From Eq. (8), one can see that the expected channel assignment delay $T_{\text{delay}}(u)$ is determined by the following four factors.



(a) s=7 (b) k=4

Fig. 6. Channel assignment delay vs. the number of total channels in the network.

(i) T_{neg}: the expected channel assignment delay increases with the increase of T_{neg}. This is due to a fact that a larger T_{neg} will lead to longer waiting time for the sender to execute *Step 3* of Fig. 3, even when none of its idle neighbor nodes shares a common channel with the sender;

(ii) $|S_C|$: the expected channel assignment delay increases with the increase of $|S_C|$, which also can be seen from the numeric results shown in Fig. 6a. With the increase of $|S_C|$, the probability that no idle neighbor node has the same assigned channel as the sender is increased. As a result, the expected channel assignment delay tends to be increased;

(iii) k: the expected channel assignment delay increases with the increase of k, which can be observed from the numeric results shown in Fig. 6a. Like the case with larger $|S_C|$, larger value of k also leads to higher probability that no idle neighbor node has the same assigned channel as the sender. As a result, the expected channel assignment delay tends to be increased;

(iv) s: the expected channel assignment delay decreases with the increase of s, which also can be seen from the numeric results shown in Fig. 6b. This is because a larger value of s results in a higher probability that an idle neighbor node shares a common channel with the sender.

3.1.4. Impact of time asynchrony on the proposed spatial assignment scheme

Here the clock drift of each node from the standard time, denoted as Δt, is assumed to be a random variable following a normal distribution with the parameter $(0, \sigma)$, where σ can be extracted from the measurement records and determined by a synchronization algorithm similar to the one used in [26].

In the proposed spatial scheme, time asynchrony may introduce collisions between neighbor nodes. For instance, after a sender node u broadcasts a control message CRE with the usage information of channel i, one of node u's neighbor nodes, say node v, will insert channel i's CBT and ADP into node v's CUT but this is done based on node u's own clock. If node v's clock is faster than node u's, node v will remove channel i usage record from its CUT even before node u actually finishes its communication on channel i. If node v immediately starts to use channel i for data transmission, a collision will occur between node u and node v on this channel. We now can calculate the expected collision probability due to time asynchrony, denoted as P_c. Note that if node v uses a different channel j to communicate after it removes channel i usage record, the collision will not occur. Therefore, P_c shall be viewed as an upper bound of the actual collision probability due to time asynchrony.

Proposition 2. For any node (say node u) in the network, due to clock asynchrony, an upper bound of the collision probability, P_c, is $0.9 s_f \sigma (\frac{1}{2})^{s_f}$, where s_f is the number of node u's neighbors whose clocks are faster than node u and σ is the variance of the clock drift of each node from the standard time.

Proof. Without loss of generality, we assume that node u's clock is set the standard time and there are s_f neighbor nodes whose clocks are faster than node u. Hence, we have

$$P_c = \int_{-\infty}^{0} -y f(y) dy \tag{9}$$

where $y = max\{\Delta t_i, 1 \leqslant i \leqslant s_f\}$, Δt_i is clock drift of node u's i-th neighbor from the standard time set by node u's clock, and $f(y)$ is the probability density function of y. Eq. (9) also can be rewritten as

$$P_c = \int_{0}^{\infty} x f(x) dx \tag{10}$$

where $x = min\{\Delta t_i, 1 \leqslant i \leqslant s_f\}$ and $f(x)$ is the probability density function of x. Since $\Delta t_1, \Delta t_2, \ldots, \Delta t_{sf}$ are independent random variables all following a normal distribution, we have

$$\begin{aligned} P_r(x \geq a) &= P_r(\Delta t_1 \geq a) P_r(\Delta t_2 \geq a) \ldots P_r(\Delta t_{sf} \geq a) \\ &= [1 - \Phi(a)]^{s_f} \end{aligned} \tag{11}$$

where $\Phi(a)$ is the cumulative distribution function of the standard normal distribution.From Eq. (11), we have

$$f(x) = s_f \phi(x)[1 - \Phi(x)]^{s_f - 1} \tag{12}$$

where $\phi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$ and $\Phi(x) = \int_{-\infty}^{x} \phi(x)dx$.

Substituting Eq. (12) into Eq. (10), we arrive at

$$P_c = \int_0^\infty s_f x \phi(x)[1 - \Phi(x)]^{s_f - 1} dx. \tag{13}$$

As for $x \geqslant 0$, we have $\Phi(x) \geqslant 1/2$ or $1 - \Phi(x) \leqslant 1/2$. Hence from Eq. (13), we can see that

$$P_c \leqslant \int_0^\infty s_f x \phi(x) \left(\frac{1}{2}\right)^{s_f - 1} dx = \frac{s_f \sigma}{\sqrt{2\pi}} \left(\frac{1}{2}\right)^{s_f - 2} \approx 0.9 s_f \sigma \left(\frac{1}{2}\right)^{s_f}. \tag{14}$$

It has been shown in [26] that the variance of 1-hop nodes is typically less than 0.09 s, i.e., $\sigma < 0.09$ s. Therefore, given a typical wireless mesh network where the number of a node's neighbors that have faster clocks ranges from 3 to 20 (i.e., $3 \leqslant s_f \leqslant 20$), according to Proposition 2, the upper bound of collision probability (P_c) shall be less than 0.03. This small probability of collision indicates that clock asynchrony has negligible impact on collision, and thus the proposed spatial scheme is resilient to time asynchrony.

3.1.5. Algorithm complexity analysis

3.1.5.1. Time complexity. For the algorithm listed in Fig. 3, the first two steps, *Step 1* and *Step 2* have the time complexity of $O(1)$. For *Step 3*, the worst case occurs when the sender fails to find an idle receiver after it attempts all the available channels. In this case, the sender needs to run this step $|S_C|$ times and therefore, *Step 3* has the time complexity of $O(|S_C|)$. The time complexity of Fig. 4 is found to be $O(1)$. Therefore, the time complexity of the proposed spatial scheme is bounded to $O(|S_C|)$.

3.1.5.2. Message complexity. In *Step 3* of the algorithm listed in Fig. 3, several rounds of message exchanges are needed for a sender node u, including (i) 1 message exchange required in line 9, (ii) Δ message exchanges (the worst case) required in line 12, where $\Delta = max\{|S_N(u)|, u \in N\}$, and (iii) $|S_C|$ message exchanges required in line 14. Node u needs to execute *Step 3* $|S_C|$ times at most. On one hand, if node u can establish a communication with a receiver at the ith iteration of *Step 3* ($1 \leqslant i \leqslant |S_C|$), then the number of message exchanges for node u is $(i - 1) + 1 + \Delta + |S_C|$. When $i = |S_C|$, the number of message exchanges reaches the maximal value of $2|S_C| + \Delta$; on the other hand, if node u cannot establish a communication with a receiver after it has attempted all the channels, the maximal number of message exchanges for node u will be $|S_C|$. Therefore, the message complexity of a sender node u is $O[max(2|S_C| + \Delta, |S_C|)] = O(|S_C| + \Delta)$. Similarly, for the algorithm listed in Fig. 4, the message complexity of a receiver node is $O(|S_C|)$. Therefore, the message complexity of the proposed spatial scheme is bounded to $O(|S_C| + \Delta)$.

3.1.5.3. Memory complexity. Each non-gateway node uses a CUT to store its neighbors' channel usage information. The maximal number of records in the CUT is Δ. As shown in Table 2, each entry in a CUT takes $\log(\Delta) + \log(|S_C|) + \varepsilon$ bits, where ε is a constant. Therefore, the memory complexity of the proposed spatial scheme is bounded to $O[\Delta\log(\Delta) + \Delta\log(|S_C|)]$.

3.2. Gateway node: a temporal scheme

3.2.1. Algorithm description

In a multi-channel environment, to utilize each channel and receive the packets from each 1-hop node, the gateway node shall cycle through channels 1, 2 ..., and $|S_C|$. To be fair, the original working interval of each channel is set to be equal, denoted as T_c (shown in Table 4). All other control messages as well as timers are summarized in Table 4.

The algorithms performed at a gateway node and at a 1-hop node are presented in Figs. 7 and 8, respectively.

In Fig. 7, if gateway node has not received a REPLY message to its RTRV message that was sent out through its currently assigned channel $C_A(u)$ after the timer T_{neg} expires, it will conclude that either none of the 1-hop nodes is listening to channel $C_A(u)$ or the 1-hop nodes that are listening to channel $C_A(u)$ actually have no packet to transmit. In this case, the gateway node will immediately switch to channel $[C(u) + 1]$ mod $|S_C|$. This proactive approach tends to increase the bandwidth utilization and thus improve network throughput.

In Fig. 8, once a 1-hop sender node u receives a RTRV message (from the gateway node), it will defer broadcasting its REPLY message after a back-off time $T(u)$ ($0 \leqslant T(u) \leqslant T_{neg}$), and $T(u)$ is calculated by:

$$T(u) = T_{neg}[\delta M_a(u)/M(u) + (1 - \delta)|S_C|N_t(u)/N_{SC}] \tag{15}$$

where δ is an adjustable parameter ($0 \leqslant \delta \leqslant 1$), N_{SC} is the total number of channels that have been traversed by the gateway node, and $N_t(u)$ is the total number of transmissions of node u. In Eq. (15), a 1-hop node with a lower memory usage ratio and a smaller number of transmissions has a higher chance to be selected as the sender. Introduction of back-off timer $T(u)$ will help prevent multiple 1-hop nodes that have been assigned to the same channel from transmitting their packets to the gateway node

Table 4
Message types and Timer types.

Message	Message description	Timer	Timer description
RTRV	Request-To-Receive message sent by gateway	T_{neg}	System timer triggered by gateway
REPLY	REPLY message sent by a 1-hop node	T_c	Original working interval of each channel
RESV	RESerVe message sent by gateway	$T(u)$	Timer triggered by a 1-hop node u
RTSW	Request-To-SWitch message sent by a 1-hop node		
STOP	STOP message sent by gateway		

```
Temporal_scheme (u: gateway node)
N_SC ← 1          // N_SC stores the total number of channels have been traversed by the gateway node
Broadcast a RTRV message on channel C_A(u), piggybacking with a value of N_SC;
Initialize a timer T_neg;
while (current clock time t_cur < T_neg)   {
    if (node u receives a REPLY message) then   {
        Cancel the timer T_neg;
        Broadcast a RESV message;
        Initialize a timer T_c;            // T_c is a timer that guarantees the fair channel utilization
        while (current clock time t_cur < T_c)   {
            if (node u receives a Request-To-SWitch (RTSW) message) then   {
                Cancel the timer T_c;
                C_A(u) ← [C_A(u)+1] mod |S_C|, N_SC ← N_SC +1;
                goto line 1;   }   }       // node u switches to channel C_A(u) reactively
        Broadcast a STOP message;
        C_A(u) ← [C_A(u)+1] mod |S_C|, N_SC ← N_SC +1;
        goto line 1;   }   }               // node u switches to channel C_A(u) proactively
C_A(u) ← [C_A(u)+1] mod |S_C|, N_SC ← N_SC +1;
goto line 1;                               // node u switches to channel C_A(u) proactively
```

Fig. 7. Temporal scheme for a gateway node.

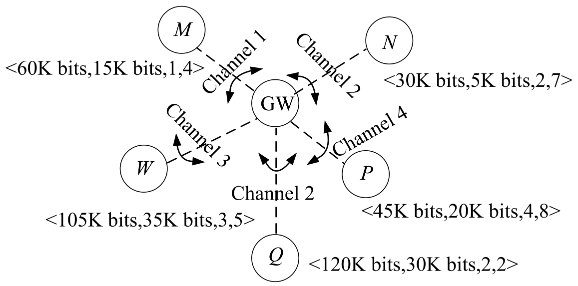
```
Temporal_scheme (u: 1-hop node)
if (node u receives a RTRV message) then   {
    if M_a(u) < M(u) then   {      // node u has data packets to transmit
        N_t(u) ← number of transmissions between node u and gateway before current time instance t;
        T(u) ← T_neg [ δ M_a(u)/M(u)) + (1−δ) |S_C|N_t(u)/N_SC ];
        Initialize the timer T(u);
        while (current clock time t_cur < T(u))   {
            if (node u receives a RESV message) then   {
                Cancel the timer T(u);
                exit;   }   }
        Broadcast a REPLY message and mark itself as a sender node on channel C_A(u);   }   }
elseif (node u receives a RESV message and node u is a sender of channel C_A(u)) then   {
    Transmit the data packets to the gateway;
    Broadcast a RTSW message;   }   // node u stops the communication with the gateway proactively
elseif (node u receives a STOP message and node u is a sender of channel C_A(u)) then
    Stop data transmission;
```

Fig. 8. Temporal scheme for a 1-hop sender node.

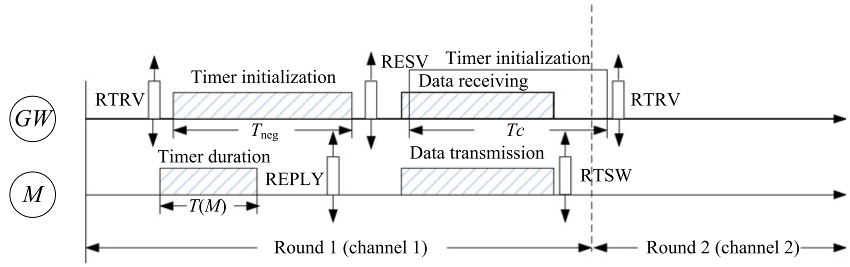
simultaneously. To minimize the algorithm overhead, T_c is set to be longer than T_{neg}.

3.2.2. An illustrative example

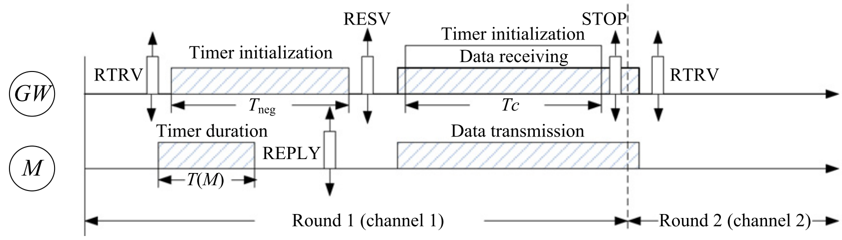
Fig. 9a illustrates a simple working example including five 1-hop nodes and one gateway node. Here any 1-hop node, say node P, is associated with (i) node P's memory size $M(P)$, (ii) node P's current available memory size $M_a(P)$, (iii) node P's current assigned channel $C_A(P)$, and (iv) number of transmissions as a sender $N_t(P)$. Assume that $|S_C| = 4$, node M stays on channel 1, nodes N and Q on channel 2, node W on channel 3, and node P on channel



(a) Logical topology of a gateway node and five 1-hop nodes



(b) Message sequence of the temporal scheme



(c) Message sequence of the temporal scheme

Fig. 9. An example with five 1-hop nodes and one gateway node to show the algorithms described in Figs. 7 and 8. Taking node M in (a) as an example. The total memory size of node M is 60 K bits, of which 15 K bits is currently available. In addition, channel 1 is assigned to node M, and M, as a sender, has transmitted 4 times to the gateway node.

4. Then the gateway node GW can receive the data packets from all the 1-hop nodes by switching to channels 1, 2, 3, and 4 periodically.

After node GW switches to channel 1, it first broadcasts a RTRV message. Meanwhile, the timer T_{neg} is also initialized (lines 1–2 of Fig. 7). Once a 1-hop node (node M in this case) receives a RTRV message, it will defer broadcasting a REPLY message to compete for the media access after a back-off time $T(M)$ (lines 1–10 of Fig. 8). When the gateway node receives a REPLY message (from node M) before T_{neg} expires, it will broadcast a RESV message to prohibit other 1-hop nodes that are also assigned to the same channel from transmitting. To limit excessively long usage of channel 1, a timer T_c which is shown in lines 4–7 of Fig. 7 is initialized. When the 1-hop sender, node M, receives the RESV message, it begins transmitting its data packets to node GW on channel 1 (lines 11–12 of Fig. 8). If node M finishes its data transmission before the timer T_c expires, it will disconnect the data communication by broadcasting a RTSW message (line 13 of Fig. 8). Upon receiving this message, the gateway node GW will switch to the next channel (channel 2 in this case) immediately (lines 9–12 of Fig. 7). The complete time lines of above message sequence for nodes GW and M are shown in Fig. 9b. On the other hand, if node M does not finish the data transmission after the timer T_c expires (i.e., node GW has not received RTSW message after the timer T_c expires), node GW will disconnect the data communication by broadcasting a STOP message and then switch to channel 2 immediately (lines 13–14 of Fig. 7). Upon receiving this message, the sender, node M, will stop its transmission (lines 14–15 of Fig. 8). In this case, the complete time lines of above message sequence for nodes GW and M are shown in Fig. 9c.

After the gateway node GW switches to channel 2, either node N or Q has its chance to access the medium, which is dependent on the timer given by Eq. (15). With $N_{SC} = 50$ and $\delta = 0.3$, we have $T(N) = 0.442T_{neg}$ and $T(Q) = 0.187T_{neg}$, which corresponds to the situation that node Q will be a sender on channel 2. Given $\delta = 0.9$, we have $T(N) = 0.206T_{neg}$ and $T(Q)=0.241T_{neg}$, which will put node N as the sender on channel 2. Similarly, node W and node P also can communicate with the gateway node GW on channels 3 and 4, respectively.

3.2.3. Algorithm complexity analysis

3.2.3.1. Time complexity. From Figs. 7 and 8, following the similar analysis method as reported in Section 3.1.5, one can see that for each channel, the time complexity of the proposed temporal scheme is bounded to $O(1)$.

3.2.3.2. Message complexity. In Fig. 7, gateway node will exchange messages as required in lines 1, 4, 6, 9, and 13. Note that gateway node cannot receive messages as mentioned in lines 9 and 13 at the same time, as these two messages are mutually exclusive. Thus, the number of messages for gateway node is $\Delta + 3$ (worst case), where $\Delta = max\{|S_N(u)|, u \in N\}$. In Fig. 8 and 1-hop sender node u will exchange messages as shown in lines 1, 7, 10, 11, 13, and 14. Similarly, the messages mentioned in lines 7 and 10, lines 13 and 14, are also mutually exclusive. In this case, the number of messages for a 1-hop count is 4 (worst case). Therefore, the message complexity of the proposed temporal scheme is bounded to $O(\Delta)$.

3.2.3.3. Memory complexity. Since gateway node and all the 1-hop sender nodes do not need to maintain their CUTs in the temporal scheme; the memory usage of the proposed temporal scheme is literally zero.

Putting the results derived from Sections 3.1.5 and 3.2.3 together, we can see that:

(i) Time complexity of the proposed joint channel assignment and routing scheme is $O(|S_C|)$
(ii) Message complexity of the proposed joint channel assignment and routing scheme is $O(|S_C| + \Delta)$.
(iii) Memory complexity of the proposed joint channel assignment and routing scheme is $O[\Delta\log(\Delta) + \Delta\log(|S_C|)]$.

4. Simulations and results

4.1. Simulation environment

In this section, we compare the performance of four multi-channel assignment schemes through extensive simulations. The first is the linear programming formulation (Eqs. (1)–(6)) to model multi-channel single-radio optimization problem MCAO, which is solved by CPLEX [27]. The second is the proposed distributed, temporal–spatial multi-channel single-radio assignment and routing scheme (TSMAR in short). The third is the distributed multi-channel single-radio protocol LCM [21]. In LCM, the interval of current control window is set to $(l+1)T_{\text{neg}}$, where l is the number of local negotiations that a sender heard in the last control window, and the interval of data window is set to the interval where a sender can send all its packets stored in its buffer to a receiver. The last is the distributed multi-channel two-radio assignment and routing algorithm ROMA [15]. TSMAR, LCM, and ROMA are simulated by the discrete event-driven network simulator NS-2 with CMU wireless extensions [28].

Here a total of 100 nodes are randomly distributed in a rectangular region with the gateway node placed at the center. Two network scenarios (with different node densities) are simulated. The first (second) scenario is created by randomly placing 100 nodes in a $500 \times 500\ \text{m}^2$ ($1000 \times 1000\ \text{m}^2$) area. The basic simulation parameters are listed in Table 5. Here 50 constant bit rate (CBR) traffic flows, over UDP, are generated from 50 randomly selected nodes. The data packet generation rate for each flow is varied to create a load variation in the network and simulations are performed for different numbers of channels. IEEE 802.11 DCF is used as the MAC protocol without any change. In addition, considering that the functions of T_{neg} and T_c are similar as those of control window duration and data window duration in protocol MMAC [20], we use the same values of T_{neg} and T_c in the simulation as MMAC. That is, the values of T_{neg} and T_c are set to 20 ms and 80 ms, respectively. The values of α, β and δ are determined by the empiricism. The simulation time is set to be 20 s, which is long enough to make sure that the gateway receives data packets from all the source nodes.

The following performance measures are used for comparison.

(1) *Aggregate network throughput*. As a measure of total network capacity, the aggregate network throughput is defined as the received bits by all the nodes per second, which is the most important performance metric to compare the efficiency of different channel assignment schemes;
(2) *Saturation network throughput.* It is the maximal throughput that a network can accommodate, i.e., the aggregate network throughput will not be larger than the saturation network throughput even if the nodes can generate more data packets. The scheme with higher saturation network throughput indicates it is more scalable to accommodate increased traffic load;
(3) *Collision ratio*. It is defined as the ratio of the number of data packets that are lost during the communication to the number of data packets that are transmitted by all the nodes. The scheme with lower collision ratio corresponds to higher reliability of the network;
(4) *Overhead ratio*. It is defined as the value $S_{\text{con}}/(S_{\text{con}} + S_{\text{dat}})$, where S_{con} denotes the total size of control messages that are transmitted over the network

Table 5
Simulation parameters.

Parameter	Value
Network region size	$500 \times 500\ \text{m}^2$, $1000 \times 1000\ \text{m}^2$
Number of nodes	100
Channel bandwidth (C)	1 M bps
Radio propagation model	Two-ray ground reflection model
Node radio transmission range (r_t)	100 m
Number of CBR traffic flows	50
Control message size	16 bytes
Data packet size	512 bytes
Node memory size	[1 M bytes, 5 M bytes]
Value of T_{neg}	20 ms
Value of T_c	80 ms
Adajustable parameters α, β and δ	0.3, 0.2, 0.6
Simulation duration	20 s

and S_{dat} denotes the total size of data packets that are transmitted over the network. The scheme with lower overhead ratio corresponds to a better one;

(5) *Packet delivery latency*. It is the time interval from the moment when a data packet is generated to the moment when the gateway node receives this data packet;

(6) *Channel usage ratio*. Given channel i, it is defined as the ratio of the numbers of channel i that is used by the algorithm to the number of all channels that are used by the algorithm for the duration of the simulation;

(7) *Memory usage ratio*. It is defined as the ratio of the size of occupied memory to size of total memory. This metric shows the influence of channel assignment scheme on the memory usage level at various nodes.

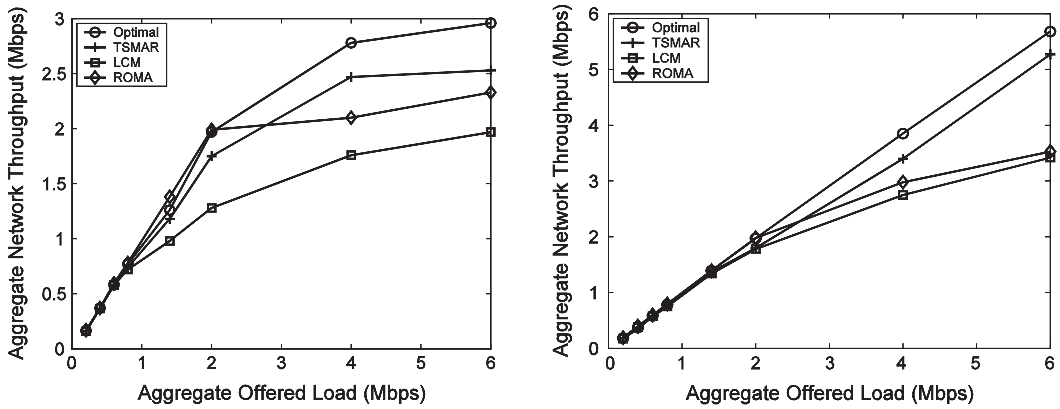
4.2. Simulation results

In our simulations, 10 different random network topologies are generated. For each topology, 50 sets of source nodes are randomly selected. Each scheme is evaluated on each set of source nodes. The simulation results are plotted using the average values derived from 500 experiments, with a 95% confidence interval. A more effective scheme is signified by the higher values in aggregate/saturation network throughput, as well as by the lower values in collision ratio and overhead ratio, packet delivery latency, and memory usage ratio, while preserving a stable channel usage ratio.

4.2.1. Aggregate network throughput

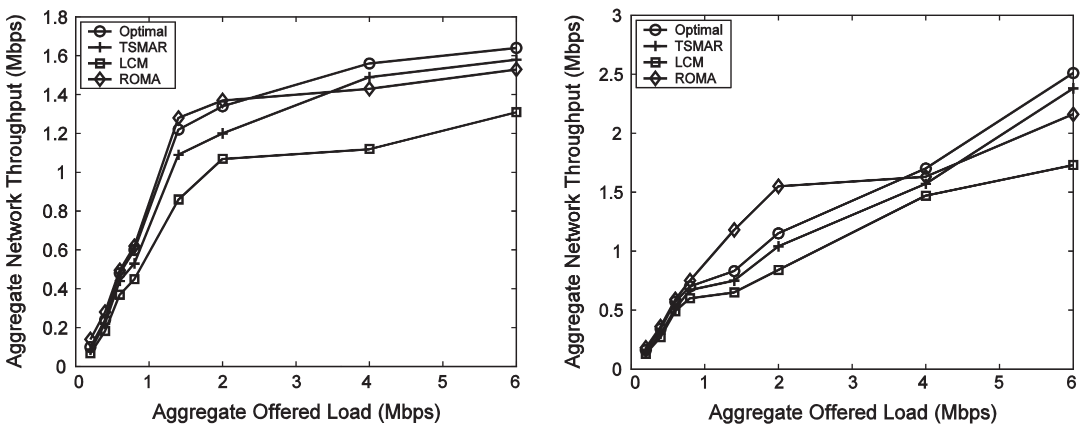
Figs. 10 and 11 plot the simulation results of aggregate network throughput vs. aggregate offered load under two network scenarios, respectively.

From Figs. 10 and 11, one can see that the difference of aggregate network throughput between the optimal solution using linear programming and the proposed TSMAR using distributed, heuristic approach is small (e.g., the ratio of the latter to the former is not less than 85.5%), which indicates TSMAR is an effective multi-channel assignment scheme. Another observation is that at low traffic loads (e.g., varying from 0.2 Mbps to 2 Mbps), ROMA generally



(a) 4 channels (b) 11 channels

Fig. 10. Aggregate network throughput vs. offered load with 1Mbps channels in a $500 \times 500\ m^2$ area.



(a) 2 channels (b) 4 channels

Fig. 11. Aggregate network throughput vs. offered load with 1Mbps channels in a $1000 \times 1000\ m^2$ area.

performs better than TSMAR and the optimal linear programming formulation, as each node in the former has actually two interfaces working simultaneously, as opposed to just one-interface in the latter two schemes. When the traffic load increases (e.g., varying from 2 Mbps to 6 Mbps), compared with the latter two schemes, the introduction of time synchronization in ROMA obliterates the benefit of having two interfaces, which results in a noticeable degradation of network throughput. Note that when the traffic load is low, having multiple channels does not bring in any additional benefit because the total traffic is less than the capacity of a single channel.

It is fair to compare TSMAR and LCM together as they both use a single-radio interface. One can see that TSMAR can achieve better network throughput than LCM in all cases, largely due to the fact that (1) for TSMAR, the control messages or data packets can be transmitted on any channel; (2) a node can switch to another channel for data communication (or control message exchange) independently. By contrast, for LCM, although the intervals of control window and data window for different cycles can be varied, they are fixed for a specific control window, which is decided by the *master node* [21]. Furthermore, LCM requires that the communication nodes keep the same channel for every data window (i.e., they cannot change the current channel until the end of current data window) even if they have finished their data transmissions. These drawbacks in LCM tend to introduce control/data window inefficiencies, resulting in a waste of bandwidth. It can be seen from Figs. 10 and 11, the aggregate network throughput of TSMAR is higher than that of LCM by up to 54.1% at medium to heavy traffic load.

In addition, from Figs. 10 and 11, it also can be seen that the aggregate network throughput increases with the increase of the number of channels. This is due to the fact that with more channels available, it becomes easier for senders to find channels to transmit, which helps cut down the waiting time. However, providing more channels alone cannot considerably improve the network throughput. Actually, for our simulation setting with 100 nodes distributed in a $500 \times 500\,\mathrm{m}^2$ or a $1000 \times 1000\,\mathrm{m}^2$ area, every node on average has roughly $100^3\pi/(500^2) \approx 12$ or $100^3\pi/(1000^2) \approx 3$ neighbors. That is, 12 (3) channels shall be sufficient to support all the neighbors transmitting simultaneously on different channels. We also note that the aggregate network throughput of all three schemes decreases with the increase of the deployment area because larger deployment area indicates that data packet may have to traverse more hops to reach the gateway node.

4.2.2. Saturation network throughput

One goal of our work is to demonstrate the performance benefit of using multiple channels in wireless mesh networks. To this end, we plot the saturation network throughputs of TSMAR and LCM in Fig. 12, with the numbers of channels varies from 4 to 13 in a $500 \times 500\,\mathrm{m}^2$ area or from 2 to 5 in a $1000 \times 1000\,\mathrm{m}^2$ area. For baseline comparison, results from single-channel IEEE 802.11 are also included.

Fig. 12a shows that the saturation throughput of both TSMAR and LCM increases nearly linearly with the number of channels $|S_C|$ going from 4 to 11 in a $500 \times 500\,\mathrm{m}^2$ area. Moreover, the proposed TSMAR scheme achieves the largest saturation throughput among all three schemes in all cases. This is largely because TSMAR does not suffer from any control period inefficiency as LCM does. Note that, when $|S_C| = 13$, the saturation throughputs of TSMAR and LCM are literally saturated (i.e., no noticeable increase as compared with the case when $|S_C| = 11$). Actually, to saturate the network throughput, the maximal number of channels that can be fully utilized in the network is bounded by the maximal number of neighbors of a node. Fig. 12b shows a similar trend for the saturation throughput vs. $|S_C|$ with nodes spread over in a larger $1000 \times 1000\,\mathrm{m}^2$ area.

4.2.3. Collision ratio

Any dynamic multi-channel protocol must ensure that the sender and the receiver are on the same channel before the communication. To achieve this, it either ensures the communication nodes switch to a pre-determined channel at a pre-determined time (e.g., LCM), or uses time synchronization to perform a channel negotiation (e.g., ROMA) process. As of TSMAR, the sender and the receiver are

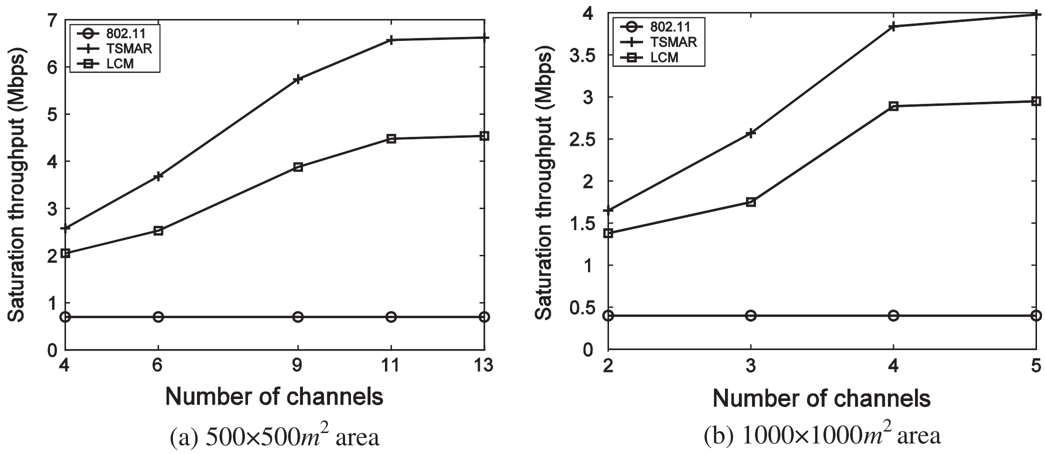


Fig. 12. Saturation throughput comparison of TSMAR and LCM vs. number of channels.

guaranteed to be on the same channel with the introduction of the back-off timers.

Fig. 13 shows the collision ratio of the three schemes applied to the nodes spread across a $500 \times 500\ m^2$ area. Compared with LCM and ROMA, TSMAR has a slightly higher collision ratio due to the following two reasons:A node updates its CUT based on the control messages (i.e., RTR and CRE) it has received (shown in Figs. 3 and 4). That is, if nodes in a TSMAR-enabled network cannot receive the latest control messages, they will maintain their CUTs with obsolete channel usage information of their neighbors. In this case, some neighboring nodes may simultaneously use the same channels, resulting in collision of data transmissions;Time asynchrony introduces collision of data transmissions. The upper bound of this collision probability was given in Proposition 2.The simulation result shows that collision ratio of TSMAR is below 6.08% in the worst case, which is still an acceptable level. On the other hand, we observe that TSMAR with the larger number of channels has a slightly higher collision ratio since the communication nodes have to switch to more channels to broadcast their control messages and thus have a higher possibility to hold obsolete information in their CUTs during the channel switching. This slight increase of collision ratio, however, is well offset by the much improved aggregate network throughput (shown in Figs. 10 and 11), thus becoming invisible by the end users. Moreover, as Fig. 13 demonstrates that the collision ratio of TSMAR can maintain the stable status with the increase of aggregate offered load, it is safe to say that the TSMAR scheme is applicable to the WMN with heavy traffic load.

4.2.4. Overhead ratio

Fig. 14 shows the overhead ratio of the three schemes applied to the nodes spread across a $500 \times 500\ m^2$ area.

On one hand, as the aggregate offered load increases, the overhead ratio of TSMAR decreases due to the fact that the total size of control messages (i.e., S_{con}) keeps same approximately; on the other hand, given the same aggregate offered load, the value of S_{con} is proportional to $|S_C|$, which has derived in Section 3.2.3. Consequently, the overhead ratio of TSMAR increases with the number of total channels. Nevertheless, the overhead ratio of TSMAR is still acceptable because the maximal overhead ratio is around 11.1%. In addition, compared with TSMAR, the overhead ratios of LCM and ROMA keep more stable. The reason is that the latter two schemes utilize dedicated channel or time synchronization for channel coordination.

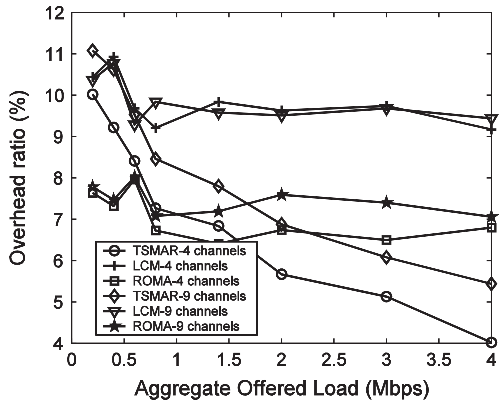


Fig. 14. Overhead ratio vs. offered load with 1Mbps channels in a $500 \times 500\ m^2$ area.

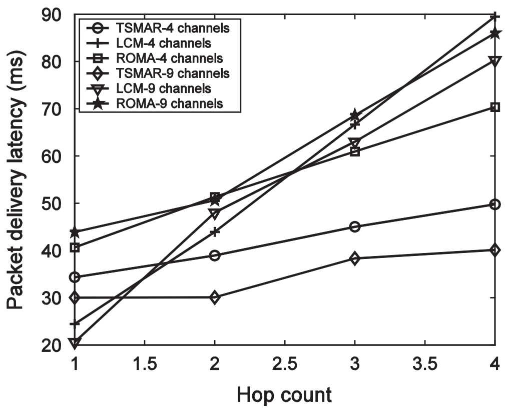


Fig. 15. Packet delivery latency vs. source node with different hop counts in a $500 \times 500\ m^2$ area.

4.2.5. Packet delivery latency

Given different numbers of channels, Fig. 15 shows the average delivery latency per packet that is generated by the source nodes with different hop counts. The packet delivery latency is the summation of time spent on channel assignment and time spent on packet routing.

From Fig. 15, one can see that the packet delivery latency is proportional to the hop count in most cases, as the source node with a larger hop count has to transmit its packets via more hops. This figure also demonstrates that the packet delivery latency of TSMAR is up to 50.02% and 44.18% less than that of LCM and ROMA, respectively. One big reason for TSMAR's much reduced latency, as shown in Fig. 15, is attributed to the fact that the senders in TSMAR do not need to wait if they can find an available

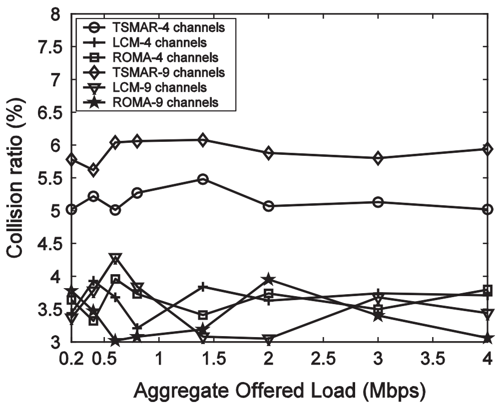
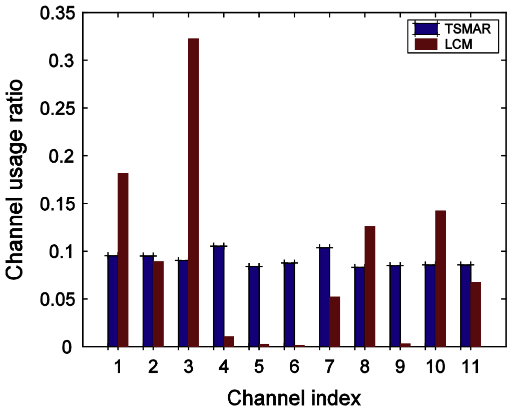
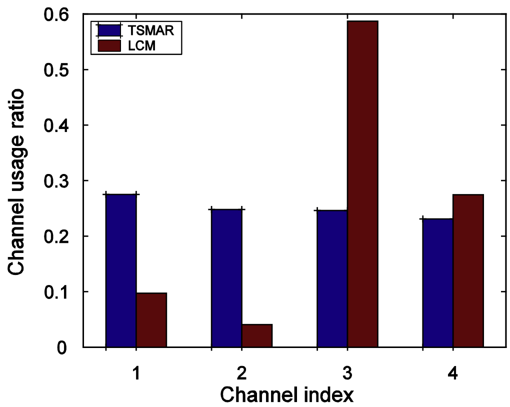


Fig. 13. Collision ratio vs. offered load with 1Mbps channels in a $500 \times 500\ m^2$ area.



(a) 11 channels in a 500×500m^2 area



(b) 4 channels in a 1000×1000m^2 area

Fig. 16. Channel usage ratio vs. channel number.

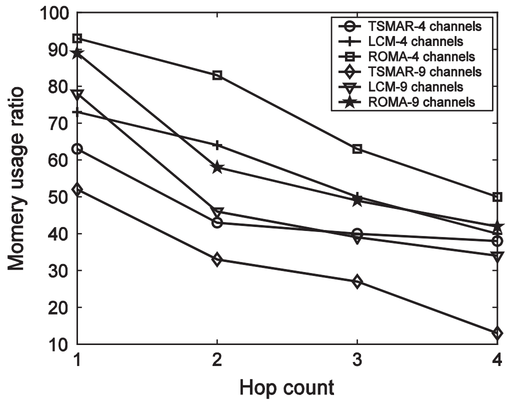


Fig. 17. Memory usage ratio vs. hop count in a 500 × 500 m^2 area.

channel. In addition, unlike LCM, routing-related information, such as hop count, is considered in TSMAR when a node needs to select a feasible path to the gateway node. Another drawback in LCM and ROMA is that some senders have to hold their transmissions until the next cycle starts, even though some channels are already available. Note that the packet delivery latency of all these three schemes will decrease with the increase of number of channels, due to more available channels and thus less waiting time incurred to those nodes that need to transmit.

Similar results are obtained for nodes spread over in a larger 1000 × 1000 m^2 area.

4.2.6. Channel usage ratio

Fig. 16 demonstrates the effectiveness of multi-channel assignment by applying TSMAR and LCM.

In Fig. 16a, one can see that, when TSMAR is applied to nodes spread across a 500 × 500 m^2 area, channel usage ratio of each channel is close to $1/|S_C| = 0.091$ where $|S_C|$ is the number of all the available channels. This result indicates that a total of eleven available channels can be fairly assigned. Similarly, each channel of TSMAR across a 1000 × 1000 m^2 area can be assigned fairly with a channel usage ratio closing to 1/5 = 0.25, as shown in Fig. 16b. Non-gateway communication node can achieve this level of fairness in terms of channel usage ratio because it checks its CUT before it selects an available channel; that is, a node selects a channel that is not being used by any of its neighbors. Gateway node can achieve fair channel usage because it uses a temporal, round-robin channel assignment scheme to traverse each channel. In contrast, the node in LCM selects channel without considering its neighbors' channel usage situation.

4.2.7. Memory usage ratio

To observe the influence of different channel assignment schemes on memory usage ratio, here the simulation duration is set to be smaller than 20 s so that data packets are still stored in a number of nodes.

The relationship between the memory usage ratio and the node hop count is shown in Fig. 17, where TSMAR exhibits a significantly lower memory usage ratio than LCM and ROMA. It also indicates that the memory usage ratio of a node with a larger hop count is less than that of a node with a smaller hop count. This is because in the latter case, not only does a node need to store the data packets generated by itself, likely it also needs to store the data packets from other nodes. In addition, it has also been observed that the memory usage ratio is inversely proportional to the number of channels. The reason is that with the increased number of channels, more nodes are allowed to transmit their data packets stored in their memories without waiting.

Similar results have been observed for the case when the nodes are distributed in a larger 1000 × 1000 m^2 area.

In summary, the performance study in this section demonstrates that the proposed scheme TSMAR, with performance close to the optimal solution, is an effective, distributed multi-channel assignment and routing scheme to solve the optimization problem MCAO (shown in Section 2.3). In addition, extensive simulation results demonstrate that TSMAR can achieve high network throughput and optimized routing path selection (in terms of low packet delivery latency and low memory usage ratio), while maintaining the fair usage of each channel. As hardware cost of TSMAR-enabled network in terms of memory usage is significantly lower than the two popular channel assignment schemes LCM and ROMA, TSMAR is particularly suitable for the many networks where nodes often have constrained hardware resources.

5. Conclusions

In this paper, a temporal–spatial multi-channel assignment and routing scheme was proposed to improve network performance in wireless mesh networks. As this proposed scheme does not require a dedicated channel for coordinating the communication, the waste of bandwidth resource tends to be considerably reduced. In addition, as time synchronization is not required in the proposed scheme, a node's idle time measured in clock cycles is decreased, leading to enhanced channel utilization. Furthermore, this proposed scheme accounts for various routing metrics to establish optimized routing paths for packet delivery from source nodes to the gateway, which also contributes to efficient bandwidth resource utilization. The complexity analysis also verifies the low computation cost and memory requirements that will incur in every wireless node, making the proposed scheme very suitable to be applied to large-scale, resource-constrained WMNs. Simulation results have confirmed that the proposed scheme can substantially improve both the aggregate and the saturation network throughputs accompanied with a slight increase of collision ratio and an acceptable overhead ratio.

Acknowledgment

The authors acknowledge the financial support provided by the U.S. Nevada NSF EPSCoR ''Nevada Infrastructure for Climate Change Science, Education, and Outreach'' (grant number EPS-0814372) and National Natural Science Foundation of China (grant number 60903047).

References

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

[2] K. Jain, J. Padhye, V.N. Padmanabhan, L. Qiu, Impact of interference on multi-hop wireless network performance, in: Proceedings of 9th ACM International Conference on Mobile Computing and Networking (MobiCom), 2003, pp. 66–80.

[3] F. Li, Y. Wang, X-Y. Li, A. Nusairat, Y. Wu, Gateway placement for throughput optimization in wireless mesh networks, Mobile Networks and Applications 13 (1-2) (2008) 198–211.

[4] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: Proceedings of 10th ACM International Conference on Mobile computing and networking (MobiCom), 2004, pp. 114–128.

[5] J. Crichigno, M.Y. Wu, W. Shu, Protocols and architectures for channel assignments in wireless mesh networks, Ad Hoc Networks 6 (7) (2008) 1051–1077.

[6] 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 24 (11) (2006) 2028–2039.

[7] Q. Pang, S.C. Liew, V.C.M. Leung, Design of an effective loss-distinguishable MAC protocol for 802. 11 WLAN, IEEE Communications Letters 9 (9) (2005) 781–783.

[8] IEEE 802.11a Standard. <standards.ieee.org/getieee802/download/802.11a-1999.pdf>.

[9] S. Avallone, I.F. Akyildiz, A channel assignment algorithm for multi-radio wireless mesh networks, Computer Communications 31 (7) (2008) 1343–1353.

[10] S. Sridhar, J. Guo, S. Jha, Channel assignment in multi-radio wireless mesh networks: a graph-theoretic approach, in: Proceedings of 1st International Conference on Communications Systems and Networks, 2009, pp. 1–10.

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

[12] M. Alicherry, R. Bhatia, E. Li, Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks, IEEE Journal on Selected Areas in Communications 24 (11) (2006) 1960–1971.

[13] X-Y. Li, A. Nusairat, Y. Wu, Y. Qi, J. Zhao, X. Chu, Y. Liu, Joint throughput optimization for wireless mesh networks, IEEE Transactions on Mobile Computing 8 (7) (2009) 895–909.

[14] P. Kyasanur, N.H. Vaidya, Routing and interface assignment in multi-channel multi-interface wireless networks, in: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), 2005, pp. 2051–2056.

[15] A. Dhananjay, H. Zhang, J. Li, L. Subramanian, Practical, Distributed Channel Assignment and Routing in Dual-radio Mesh Networks, ACM Special Interest Group on Data Communication (SIGCOMM), 2009, pp. 99–110.

[16] S-H. Kim, Y-J. Suh, A distributed channel assignment protocol for rate separation in wireless mesh networks, Computer Communications 33 (11) (2010) 1281–1295.

[17] P. Kyasanur, N.H. Vaidya, Capacity of multi-channel wireless networks: impact of number of channels and interfaces, in: Proceedings of 11th ACM Annual International Conference on Mobile computing and Networking (MobiCom), 2005, pp. 43–57.

[18] E. Aryafar, O. Gurewitz, and E. W. Knightly, Distance-1 constrained channel assignment in single radio wireless mesh networks, IEEE Annual International Conference on Computer Communications (INFOCOM), 2008, 1436-1444.

[19] P. Bahl, R. Chandra, J. Dunagan, SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks, in: Proceedings of 5th ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc), 2004, pp. 216–230.

[20] J. So, N.H. Vaidya, Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver, in: Proceedings of 5th ACM International Symposium on Mobile ad hoc networking and Computing (MobiHoc), 2004, pp. 222–233.

[21] R. Maheshwari, H. Gupta, S.R. Das, Multichannel MAC protocol for wireless networks, in: Proceedings of IEEE Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2006, pp. 393–401.

[22] J. Shi, T. Salonidis, E.W. Knightly, Starvation mitigation through multi-channel coordination in CSMA multi-hop wireless networks, in; Proceedings of 7th ACM International Symposium on Mobile ad hoc networking and Computing (MobiHoc), 2006, pp. 214–225.

[23] Y. Jin, J-Y. Jo, M. Yang, Y. Kim, Y. Jiang,J. Gowens, HTSMA: a hybrid temporal–spatial multi-channel assignment scheme in heterogeneous wireless mesh networks, in: IEEE Global Communications Conference (GLOBECOM), 2009.

[24] R. Huang, S. Kim, C. Zhang, Y. Fang, Exploiting the capacity of multichannel multiradio wireless mesh networks, IEEE Transactions on Vehicular Technology 58 (9) (2009) 5037–5047.

[25] M. Kordialam, T. Nandagopal, The effect of interference on the capacity of multi-hop wireless networks, IEEE Symposium on Information Theory (2004) 470.

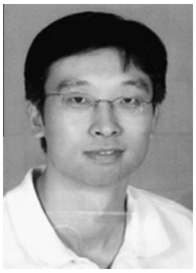
[26] Q. Gao, K.J. Blow, D.J. Holding, Simple algorithm for improving time synchronization in wireless sensor networks, Electronics Letters 40 (14) (2004).

[27] ILOG CPLEX Mathematical Programming Optimizers. <http://www.ilog.com/products/cplex/>.

[28] S. McCanne and S. Floyd, Network simulator ns-2, 1997. <http://www.isi.edu/nsnam/ns>.

Yan Jin, received his BS, MS and PhD degree in Computer Science from Harbin Institute of Technology, Harbin, China, in 2001, 2003 and 2007, respectively. He was a research fellow in Department of Electrical Engineering at University of Nevada, Las Vegas, USA from 2007-2010. From 2011, he joined Institute of Computer Technology, Chinese Academy of Sciences as a Research Associate. His research interests include network security, ad hoc networks, wireless sensor networks and wireless mesh networks.

Weiping Wang, received his PhD degree in Computer Science from Harbin Institute of Technology, Harbin, China, in 2007. He is an associate research fellow in Institute of Computer Technology, Chinese Academy of Sciences. His research interests include database and data stream processing.



Yingtao Jiang is an associate professor of Electrical and Computer Engineering at University of Nevada, Las Vegas, USA. Dr. Jiang received his Ph.D. in Computer Science from the University of Texas at Dallas. He is the author or co-author of over 70 technical publications. He is a member of IEEE. His research fields include system on Chip designs, reconfigurable computing, biomedical signal processing and medical informatics, and nano technology.

Mei Yang is an associate professor of Department of Electrical and Computer Engineering at University of Nevada, Las Vegas, USA. She received her Ph. D. in Computer Science from the University of Texas at Dallas in Aug. 2003. Her research interests include computer architectures, networking, and embedded systems. She has published over 60 journal and conference papers in these areas. She has served as PI and Co-PI for a variety of interdisciplinary projects on network-on-chips, multi-core programming, reconfigurable computing, wireless networks, sensor networks, and network security from NSF, UNLV RDA, US Air Force, NASA, UNLV NIA, and Microsoft Research.