Computer Science and Information Systems 14(3):661–684 https://doi.org/10.2298/CSIS170124022Z

An Optimization Scheme for Routing and Scheduling of Concurrent User Requests in Wireless Mesh Networks

Zhanmao Cao[1], Chase Q. Wu[2], and Mark L. Berry[2]

[1] Department of Computer Science, South China Normal University
Guangzhou, Guangdong 510631, China
caozhanmao@m.scnu.edu.cn
[2] Department of Computer Science, New Jersey Institute of Technology
Newark, New Jersey 07102, USA
{chase.wu, mlb32}@njit.edu

Abstract. Multiple-radio multiple-channel (MRMC) wireless mesh networks (W-MNs) have been increasingly used to construct the wireless backbone infrastructure for ubiquitous Internet access. These networks often face a challenge to satisfy multiple concurrent user requests for data transfers between different source-destination pairs with various performance requirements. We construct analytical network models and formulate such multi-pair data transfers as a rigorous optimization problem. We propose an optimization scheme for cooperative routing and scheduling together with channel assignment to establish a network path for each request through the selection of appropriate link patterns. The performance superiority of the proposed optimization scheme over existing methods is illustrated by simulation-based experiments in various types of mesh networks.

Keywords: multi-pair paths, compatible paths, multi-radio multi-channel, wireless mesh networks.

1. Introduction

The number of mobile smart terminals is exponentially increasing over years, so is the users' desire for any-where any-time access to the Internet, even in remote rural areas. Recently, Multi-radio multi-channel (MRMC) wireless mesh networks (WMNs) have emerged as a promising solution to provide convenient and ubiquitous broadband access to the Internet. In MRMC WMNs, as router nodes are equipped with multiple interfaces, they may operate in the mode of multiple input multiple output (MIMO). In fact, MRMC represents the main features of the current wireless network infrastructure, where a node interface typically communicates in a dual mode with an omni-directional antenna. Note that a radio may be viewed as the active status of two interfaces on two neighbor nodes over a common channel of wireless media.

MRMC WMNs have brought several important benefits. First of all, they provide significantly more capacity with higher energy efficiency than their predecessors [19]. Secondly, WMNs offer unprecedented flexibility and convenience to expand the covering area by relaying packets hop-by-hop without the support of BS in the mesh mode [6]. Thirdly, the WMN topology remains relatively stable because the nodes are almost static, hence ensuring Quality of Service (QoS) in disparate environments such as a building or even a smart city. WMNs also enable fast or temporary deployment, which is critical in emergency situations [1].

MRMC WMNs often face a challenge to satisfy multiple concurrent user requests for data transfers between different source-destination pairs (s_i, d_i), $i = 1, \cdots, \rho$, with various performance requirements. Typical examples include a request from an FTP user for transferring data of a certain size z_i from s_i to d_i or any other file transfer request. The multi-pair routing (MPR) problem in WMNs, referred to as WMPR, has a critical impact on the QoS delivered to end users and the utilization of network devices deployed by service providers, especially when resources are limited by a finite number d of antennas and a finite number $|\Omega|$ of orthogonal channels in a given WMN. For illustration purposes, we provide in Fig. 1 an example with four-pair routing, i.e., (A, D), (B, J), (C, F), and (I, H).

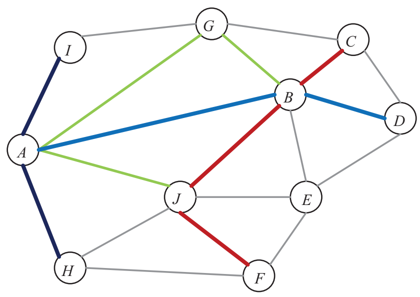


Fig. 1. An example of four compatible paths with joint nodes.

It is important to maximize the utilization of network resources. However, this problem becomes more challenging due to the interference of wireless media. In fact, the approaches to multiple pair shortest path (MPSP) widely adopted in wired networks are not suitable for wireless networks because the wireless radio interference makes this problem a discrete combinatorial one in nature [25]. In existing research, some special constraints are considered, such as edge-disjoint, minimum edge-congestion, or shortest path [4, 27]. A good scheme is needed to solve WMPR for maximum utilization of network (MUN) resources.

WMPR is both practically important and theoretically challenging. Note that each source-destination pair requires a data transfer path, and multi-pair paths may cause interferences to each other. One main goal is to carefully route multiple user requests via different paths and design an efficient scheduling scheme that allows multiple source-destination pairs to simultaneously transmit data packets over their own paths in a cooperative way. Unfortunately, The routing problem of multiple pair paths (MPP) still remains largely unexplored. Even two simplified versions of MPR, multiple pair concurrent paths (MPCP) and multiple pair shortest paths (MPSP), have not been well explored.

A good routing and scheduling scheme should aim to set up as many concurrently active paths as possible, and the links of those paths can be active simultaneously without interferences. This is important to many application scenarios that generate multi-pair data traffic in real time.

Our contributions in this work are two-fold: construct rigorous models to define WMPR, and design efficient algorithms to solve WMPR.

– Network Modeling: We construct analytical network models and formulate WMPR as a rigorous optimization problem to minimize turnaround time under various constraints.
– Cooperative Routing and Scheduling: We design a cooperative routing and scheduling scheme that dispatches multiple user requests along concurrent *compatible paths* without interferences.

The rest of the paper is organized as follows. Section 2 provides a survey of related work. Section 3 constructs network models and formulates the problem. Section 4 designs a cooperative routing and scheduling scheme. Section 5 conducts simulations in triangular meshes and random meshes for performance evaluation.

2. Related Work

As MPR has not been extensively investigated in wireless meshes, we trace several lines of research efforts in wired networks, graph theory, and transportation research.

The MPSP problem has been widely studied in various contexts. Wang *et al.* developed a DLU approach to dense digraph flight scheduling, which is similar to LU decomposition in Carr*é*s algorithm [28]. Their scheme is an algebraic matrix compared with a label-setting method and an LP-based technique.

A weighted digraph problem considers a directed graph with a set of rate demands specified on each source-destination pair $\{s_i, d_i\}$ [16]. Andrews *et al.* designed an almost-tight approximation algorithm for the directed congestion minimization problem. They chose one directed path for every pair (s_i, d_i) to minimize the maximum congestion [5]. However, their work is focused on the theoretical aspect of the problem, without considering the interferences and the limit on channel allocation (CA) in WMNs. Nevertheless, their work at least provided an analysis of the problem's computational complexity and proved the NP-completeness of the maximum utilization problem of MPR. Note that some traffic flows may have joint nodes or even common edges in the network topology.

There exist several research efforts in maximizing the utilization of wireless resources by optimizing throughput or capacity in WMNs. Alicherry *et al.* formulated CA and routing into LP by considering the characteristics of interference, the number of channels, and the number of node radios [2]. Giannoulis *et al.* proposed an iterative method to optimize congestion control by considering CA and traffic distribution [15]. They claimed that the problem is still NP-hard, even in a simpler combinatorial case on CA of MPR in multi-radio networks with a given set of rate demands. After decomposing congestion control into two stages, they formulated the MRMC congest control problem as MRMC-CC, and their optimization method considered CA between multiple pairs only, not for MPCP.

MPP in wired networks was discussed with constraints of minimum edge-congestion or maximum utility of networks [4, 5]. As a necessary step of WMPR in wireless networks, the MPP is even more complex because more constraints have to be considered for optimization of scheduling, routing, CA, and interference avoidance [25]. Even a simple combinatorial problem involving only one aspect may not have an exact optimal solution. For example, CA to meet a given set of traffic rate demands is NP-hard [24].

We would like to point out that WMPP is different from the traditional disjoint multi-path problem [27], which is focused on finding concurrent paths between one source-destination pair to improve transmission speed and reliability. Different from "multi-path" in [3] or "multipath" in [18], WMPP sets up multiple concurrent compatible paths, each for a different pair of nodes.

Traditional shortest path algorithms are not adequate to solve the problem under study. Even if we do not take into account CA and interference factors, the multiple pair shortest paths (MPSP) problem has already been proved to be NP-hard for edge congestion minimization [4]. Furthermore, these conclusions are only based on a subproblem space or a simplified case of MPP, not yet considering all aspects. For example, the model by Schumacher *et al.* does not take interference into account. In WMNs, MPCP is highly related to interference, routing, link scheduling as well as CA. Consequently, wireless MPCP (WMPCP) is at least as difficult as MPCP in wired networks. On the other hand, MPSP is NP-complete according to Karp [17]. Even only considering CA for real-time data flows, the problem is NP-complete, because CA can be reduced to the 3-partition problem [7]. The problem to perform joint scheduling and routing to achieve maximum utilization of network (MUN) resources is also NP-complete.

As discussed above, WMPR is a challenging problem and has not been thoroughly investigated. In this paper, we tackle this problem considering wireless network resources for minimum turnaround time.

3. Cost Models for WMPR

WMPR aims to achieve the maximum utilization of wireless resources or the minimum turnaround time for user requests. We consider an almost static network topology since routers are always pre-deployed and almost fixed during their operation. To facilitate a rigorous formulation of the problem, we provide below some preliminaries and notations for both the models and the algorithm to be designed.

3.1. Preliminaries

We consider an MRMC WMN structure Γ with a set V of static routers. Each router is equipped with ξ interfaces. There are q orthogonal channels $\{c_1, c_2, \cdots, c_q\}$ that are globally available, each of which has a bandwidth ϖ_i, $i = 1, 2, \cdots, q$. A router node u transmits data to its neighbor node v over a specific channel c_i. The communication link between them is denoted by a directed edge $u\overrightarrow{c_i}v$, or simply $l_{c_i}^{(u,v)}$.

Interference is an inherent nature of wireless networks. At any particular point of time, any node is allowed to send or receive data over a channel only if there is no conflict with other working nodes. Various cases of interference are discussed and categorized in [11]. With initial selected links, let S_C denote the set of possible candidate sender nodes, and let R_C denote the set of possible candidate receiver nodes. The conditions to avoid wireless interference are given in Table 1.

Table 1. Conditions for simultaneous links over one channel.

Three classes of neighbor pairs		Link interference-free conditions	
End nodes from S_C	End nodes from R_C	Necessary Condition	Sufficient Condition
S_i, S_j		$\forall i \neq j, d(S_i, S_j) \geq 2$ (1)	
	R_i, R_j	$\forall i \neq j, d(R_i, R_j) \geq 1$ (2)	(1) $\wedge$ (2) $\wedge$ (3)
S_i	R_j	$\forall i \neq j, d(S_i, R_j) > 1$ (3)	

We consider a set of traffic requests $\Lambda = \{\lambda_i | i = 1, \cdots, \rho\}$, where $\lambda_i = \{(s_i, d_i), z_i\}$, and z_i denotes the size of data to be transferred as in an FTP or any other file transfer request. In some other contexts, an explicit bandwidth may be requested.

A general solution to WMPR first computes a network path to transmit data packets of each traffic request from its source node to its destination node, followed by scheduling and CA. A network path is often defined as a finite hop-by-hop node sequence for packet forwarding. Any two nodes with a direct link in the sequence are considered as neighbors. For a pair (s_i, d_i), its network path is in the form $p_{(s_i,d_i)} = \{s_i, v_{i_1}, \cdots, v_{i_{h^i-1}}, d_i\}$ of length h^i. Given a routing path for λ_i, scheduling is to assign a transmitting time segment to each component link on the computed path, and CA is to assign a channel to each selected link according to the schedule. The scheduling turnaround for path $p_{(s_i,d_i)}$ is illustrated in Fig. 2.

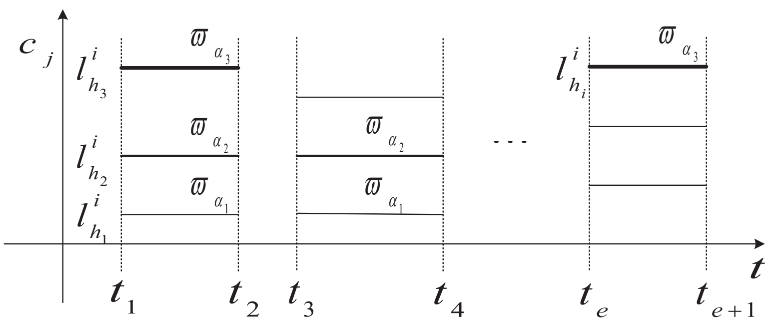


Fig. 2. An example of path scheduling over multiple channels.

Suppose that all user traffic requests are completed after total θ time segments. The s-th time segment has a duration of $t_{s+1} - t_s$. The traffic requests, scheduled links, and link bandwidths have the following relation:

$$\sum_{i=1}^{\rho} z_i \cdot h^i = \sum_{s=1}^{T} \sum_{i=1}^{\rho} \sum_{h=1}^{h^i} l^{i,h} \cdot \varpi_{i_h} \cdot (t_{s+1} - t_s), \tag{1}$$

where $l^{i,h} \cdot \varpi_{i_h}$ denotes the link bandwidth of the h-th hop of path $p_{(s_i,d_i)}$. Note that link $l^{i,h}$ has two statuses: one is active or scheduled, where $l^{i,h} = 1$; and the other is idle or unscheduled, where $l^{i,h} = 0$. An active link contains several critical parameters: 1) a pair of sender and receiver, 2) the channel it operates over, and 3) the interference-free relation.

If $l^{i,h}$ is assigned with a channel c, we use operator $c()$ to denote the CA as $c(l^{i,h})$. If $l^{i,h}$ gets channel c, we denote it as $l_c^{i,h}$. If $l_c^{i,h}$ is scheduled at time t, we set $l_{c,t}^{i,h} = 1$; otherwise, $l_{c,t}^{i,h} = 0$. For simplicity, we use $l_{c,t}^{i,h}$ to represent the assigned channel information c, scheduling time slot t, and the hop h of p_i.

For convenience, we tabulate the notations in Table 2.

Table 2. List of symbols and notations

ρ	The number of pairs
(s_i, d_i)	The i^{th} source-destination pair
Ω	The set of available orthogonal channels
c_i	The i^{th} channel in Ω
ϖ_i	The bandwidth of channel c_i
q	$q = \|\Omega\|$
ξ	The number of node interfaces
$p_{(s_i,d_i)}$	The selected path for (s_i, d_i)
θ	The total time segments needed
α	Parameter to set the number of interfaces
β	Parameter to set the number of channels
$l^{i,h}$	The h-th hop or link of $p_{(s_i,d_i)}$
$l_c^{i,h}$	The link $l^{i,h}$ using channel c
V	The set of nodes in the WMN topology
d_v	The number of interfaces equipped on node $v \in V$
N_v	The set of node v's neighbors
E	A set of neighbor pairs among V
$G = (V, E)$	A connected network graph for a WMN
$\widetilde{RSC}$	A joint scheme for routing, scheduling and CA
λ_i	The traffic request of (s_i, d_i): $\lambda_i = \{(s_i, d_i), z_i\}$
Λ	The set of traffic requests of multiple pairs $\{\lambda_i \| i = 1, \cdots, \rho\}$
f_{h,c_j}^i	The flow rate of the h-th hop of $p_{(s_i,d_i)}$ over channel c_j
T	The time period for updating the WMN structure Γ

3.2. WMPR: Multi-Pair Routing in WMNs

Given an MRMC WMN structure Γ, our goal is to maximize the number of communication paths that can be activated simultaneously to minimize the turnaround time for a given user request set Λ.

An Overview Multiple pair paths (MPP) in WMNs may have node intersections as in real-time video communications [13]. For example, in Fig. 1, node B is the intersection of two paths for two pairs C to F and A to D.

Definition 1 *Γ:*
$\Gamma = \{G, I, \Omega, \xi, \{(s_i, d_i)\}\}$*, where:*

- *$G = (V, E)$ is a mesh graph, while $|V|$ denotes the number of vertices and $|E|$ denotes the number of edges. Both $|V|$ and $|E|$ are constant integers.*
- *I is a relation for wireless interference awareness between vertices of G.*
- *$\Omega = \{c_1, c_2, ..., c_q\}$ is a set of available orthogonal channels for G. c_i has bandwidth ϖ_i.*
- *ξ is a list of integers standing for the numbers of node interfaces. By default, ξ is a constant.*
- *$\{(s_i, d_i)\}, i = 1$ to ρ is a set of multiple source-destination pairs.*
- *$\Lambda = \{\lambda_i\} = \{(s_i, d_i), z_i\}$ is the list of traffic queues corresponding to $\{(s_i, d_i)\}$ list, where $z_i > 0$.*

The substructure $\{G, I, \Omega, \xi\}$ includes the WMN infrastructure, i.e. the resources of the WMN. Meanwhile, $\{(s_i, d_i)\}$ represents multiple pairs with traffic requests $z_i > 0$, and $\rho = |\{(s_i, d_i)\}|$.

Suppose that in a given Γ, there are $|\Omega|$ channels, each channel c_i has bandwidth ϖ_{c_i}, and each router node $v \in V$ is equipped with d interfaces. We consider a set of requests $\Lambda = \{\lambda_i | i = 1, \cdots, \rho\}$, where $\lambda_i = \{(s_i, d_i), z_i\}$ and $z_i > 0$. For each request $\{\lambda_i\}$, to transmit data, we need to consider the following: find a fixed route/path $p_{(s_i,d_i)}$ for each pair, find a cooperative schedule for the links of multi-pair paths without interference, and assign channels to the scheduled links. We denote a joint scheme of these three operations as $\widetilde{RSC}$.

WMPR aims to achieve optimal joint $\widetilde{RSC}$ on routing, scheduling, and CA in a given mesh network Γ. The objectives are to minimize the turnaround time of user requests Λ, and maximize the utilization of wireless resources in serving the data transfers between multiple source-destination pairs $\{(s_i, d_i)\}$ at time t.

Our discussion is facilitated by Cartesian product of graphs (CPG), in which, each orthogonal channel is an independent virtual layer and an MRMC node is a collection of multiple fully connected identity nodes [11], as illustrated in Fig. 3.

Problem Formulation Nodes in broadband WMNs are generally equipped with multiple interfaces. The number of links that each node can use is limited by the number of interfaces and the number of available channels. We use c_i to denote any available channel, and N_v to denote the set of all neighbors of node v. Again, $l_{c_i}^{(v_1,v)}$ denotes a link from v_1 to v over channel c_i. We have the following constraint on the number of links involving node v (including both incoming and outgoing links of v):

$$\sum_{i \neq j}^{|\Omega|} \sum_{v_1, v_2 \in N_v} l_{c_i}^{(v_1,v)} + l_{c_j}^{(v,v_2)} \leq d_v, \forall v \in V, \tag{2}$$

where Ω denotes the set of channels, and d_v is the number of interfaces equipped on node v. We consider several aspects collectively: network topology, node interface, wireless interference, path selection, and CA. At time t, the channels used by node v form a true subset of Ω. $\forall v_1, v_2 \in N_v$, for links $l_{c_{i_1}}^{(v_1,v)}$ and $l_{c_{i_2}}^{(v,v_2)}$, $c_{i_1} \neq c_{i_2}$. Meanwhile, for links $l_{c_{i_1}}^{(v,v_1)}$ and $l_{c_{i_2}}^{(v,v_2)}$, if $v_1 \neq v_2$, then $c_{i_1} \neq c_{i_2}$. Similarly, for links $l_{c_{i_1}}^{(v_1,v)}$ and $l_{c_{i_2}}^{(v_2,v)}$, if $v_1 \neq v_2$, then $c_{i_1} \neq c_{i_2}$. Intuitively, any two links that share a common node v can be active simultaneously only over different channels. Also, the links of neighbor nodes on the same channel must satisfy the interference-free conditions.

The global maximum utilization of network resources requires scheduling as many links as possible in Γ. It is reasonable to maximize the number of links on multi-pair paths. If a schedule Π that achieves a maximum number of links is not a schedule for maximum utilization of networks (MUN), then there must exist another schedule that achieves MUN. Suppose that all links are of the same capacity. There must be another schedule Π' with more links. The problem should satisfy various constraints, such as multiple traffic requests, node interfaces, and free channels. Let ϖ_j denote the bandwidth of channel c_j. The waiting time for service on path $p_{(s_i,d_i)}$ is recorded as t_i. From a global perspective, to minimize the total waiting time, we should minimize the turnaround time.

The scheduled path of a pair (s_i, d_i) is denoted as $p^{sc}_{(s_i,d_i)}$. The turnaround time of $p_{(s_i,d_i)}$ is denoted as T^i. Let $\psi = |\Omega|$. The optimal WMPR problem is defined as follows:

$$
\begin{aligned}
&\min \max\{T^i | i = 1, \cdots, \rho\} \\
&s.t. \\
&\quad l_c^{i,h} = \begin{cases} 1, \text{active} \\ 0, \text{inactive} \end{cases} \\
&\quad \sum_{c_i \neq c_j}^{\Omega} \sum_{v_1,v_2 \in N_v} l_{c_i}^{(v_1,v)} + l_{c_j}^{(v,v_2)} \leq \xi, \forall v \in V; \\
&\quad f_{c_j}^{i,h} \leq l_{c_j}^{i,h} \times \varpi_j, \text{for } j = 1 \; to \; \psi; \\
&\quad \forall i, \forall h, \text{at } t, \; \{l_{c_j,t}^{i,h}\} \propto I; \\
&\quad \xi \neq 0; \\
&\quad \Omega \neq \emptyset.
\end{aligned}
\tag{3}
$$

WMPR does not have a polynomial-time exact optimal solution, even in a simple case with one single objective such as routing, scheduling, or CA. For scheduling, it has been proved to be NP-hard to determine an optimal link schedule in multi-hop radio networks [23], even if CA is not considered. For optimal scheduling, link scheduling can be converted into the edge coloring problem, which has been shown to be NP-complete [26]. For CA, which is a extensively studied problem, has been proved to be NP-hard [20]. Even a constrained version, which is a coloring problem, has been proved to be NP-complete [22]. As mentioned above, a simplified subproblem of WMPR is NP-complete. WMPR for MUN is NP-hard as proved by Schumacher *et al.*, when wireless interference is not taken into account. Without interference, it becomes a subproblem space of Γ, where $I = \emptyset$. Actually, by combining all major aspects of interference, CA, routing and scheduling over MRMC, the general problem is far more complex than the above cases that consider only one aspect.

The network topology puts a limit on the number of path options for a pair, while path selection chooses one that is interference-free with other existing paths. For example, in the channel layered virtual model in Fig. 3, the two paths share node B on different orthogonal channels. In a channel related planar mesh, the two paths successfully transmit through B's diversified identities via channels.

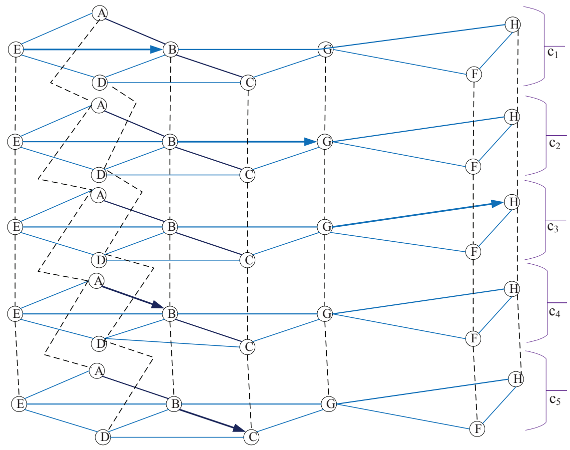


Fig. 3. Two compatible paths in CPG model.

Meanwhile, multiple pairs in a WMN fall in a situation of communication requests in an MRMC WMN defined virtual structure CPG. Note that all links of those active paths, which are simultaneously transmitting data at time t, form a substructure in CPG. We obtain a structure Γ by combining CPG and multiple pairs.

3.3. Compatible Paths

A larger number of concurrently scheduled paths for multiple pairs are able to transmit more data. Hence, it is reasonable to find more links in each channel to combine together for more paths. This idea was proposed for maximum utility of network resources [12]. However, our earlier experiments show that it is prohibitively expensive to find all interference-free link patterns in an arbitrary network topology of size beyond $|V| = 64$.

In Cartesian product of graphs (CPG), a directed path $p_{(s_i,d_i)}$ is called an *active path* at time t, if every component link is active on a distinct interference-free channel [9]. Since the CPG model maps orthogonal channels to corresponding planar meshes, a practical way is to decompose each path into links over different channel layers and choose interference-free links with maximum match to multiple pair paths.

With a minimum number of channels for each path, a good schedule can certainly lead to more active paths. To avoid the situation where several arbitrarily activated paths hog up all resources, the paths should be arranged to match the link patterns in a given mesh. Since link patterns are derived from interference-free links, we may combine links and compatible paths together for optimal performance, by searching for a link pattern from a certain link of a $p_{(s_i,d_i)}$ over interference-free channels. The link pattern search collects as many links as possible from the selected paths.

At time t, given multiple pairs $(s_i, d_i), i = 1, 2, \cdots, \rho$, and $p_{(s_i,d_i)}$, there must be some paths that can be scheduled concurrently with proper CA, i.e. without interferences and conflicts on the node interface with other existing links. Such paths are called *compatible paths*.

A major characteristic of compatible paths is the concurrent coexistence without conflicts. Each of the compatible paths is an independent packet transmission pipe for (s_i, d_i), $i = 1, 2, \cdots, \rho$. A node equipped with multiple transceivers receives data over one channel, while sending data through another interface over a different channel to guarantee co-existence.

The problem of finding the largest number of compatible paths is at least as hard as the joint path minimum edge congestion MPP, which is NP-hard [4]. With all mixed factors of WMN, such as multiple source-destination pairs, interference along a path, interference between paths, interface limits of those shared nodes, CA, node free interfaces and free channels, as well as the optimal model, we need to consider all properties in a combinatorial way. Path realization is no longer just a path selection problem in an undirected graph, because the interference and CA must be accounted explicitly, compared to the joint path minimum edge congestion MPP.

Free-channel is a true subset of the available channels and it is node related. To a mesh, the available channels of Ω refer to a list of channels that mesh nodes may choose to operate over. In a later stage, to a specific node, the free-channel denotes those channels without a conflict with the already existing assignments so far. For each node, we need to distinguish the currently feasible channels for each node from those initial available channels. The current state includes neighbor node channel settings, channels of the links surrounding the node, impacts from the channels in related paths. If two neighbor nodes have some common free channels, then they can set up a link on one of the free channels at time slot t.

The compatible paths of different pairs can be roughly classified into three classes according to the number of common nodes: i) parallel, if two paths do not share any node; ii) crossing, if two paths share one node; iii) edge-sharing, if two paths share at least one common edge (two or more common nodes). The third case may consume edge resources rapidly and generate a hole of the mesh.

To find compatible paths for multiple source-destination pairs $(s_i, d_i), i = 1, 2, \cdots, \rho$, one needs to consider interference-free constraint, number of node interfaces, as well as orthogonal channels. The link interference-free condition is considered in the sender-receiver distance relation with the node identities or coordinates [9]. Meanwhile, since parallel paths are rare, we pay more attention to the other two cases.

Generally, a common node shared by $\tilde{a}$ paths at time slot t must have more than $2\tilde{a}$ interfaces for operating on $2\tilde{a}$ orthogonal channels. For example, the path $\{A, B, D\}$ for (A, D) shares B with path $\{E, B, G\}$ for (E, G) in Fig. 1. The two paths can not be simultaneously active if they can not satisfy both of these requirements: B has at least 4 interfaces and B has at least 4 free orthogonal channels.

Two Paths with One Overlapped Node In Fig. 1, the shared node B should be equipped with more interfaces than other nodes on the two paths.

The distance matrix M records the shortest path distance of a MRMC mesh. M is essential for further routing scheme, such as the shortest path diversity routing. As an example, the distance matrix M of Fig. 1 is provided in Table 3.

Table 3. The shortest distance matrix of the pairs in Fig. 1.

From To	A	C	D	F
A	0	2	2	2
C	2	0	1	3
D	2	1	0	2
F	2	3	2	0

Suppose that in Fig. 1, a link that makes up one hop of a path is assigned weight 1 (which stands for one-hop transmission). A shortest path of a certain pair can be expressed as a finite ordered node sequence. Paths selected to realize multiple pairs can be verified with the shortest distance matrix. Note that the shortest path is not unique in some cases. For example, in Fig. 1, path C to F can be $p_{(C,F)} = \{C, B, E, F\}$ or $p_{(C,F)} = \{C, D, Q, F\}$.

The overlapped nodes of multi-pair paths may change due to path diversities. In Fig. 1, according to the two paths selected for C to F, the specific intersection node varies. Table 4 provides an example for the cases in Fig. 1. Note that $\{A, E, Q, D\}$ is not a shortest path from A to D.

Table 4. Joint nodes vary with path variations in Fig. 1.

Shared node	Two joint paths involved
B	$\{A, B, D\} \wedge \{C, B, E, F\}$
D	$\{A, B, D\} \wedge \{C, D, Q, F\}$
B	$\{A, B, D\} \wedge \{C, B, Q, F\}$
E	$\{A, E, Q, D\} \wedge \{C, B, E, F\}$

Conflicts Between Paths The number of paths is largely limited by wireless interferences and node interfaces. The paths intersecting with each other compete for the resources of overlapped nodes. If some links of a path cannot be activated because of channel and interface restriction, they may be set to time $t + 1$, and so forth, which is similar to the idea of TDMA.

As it is impossible to compute an optimal WMPP in an exact way, an alternative way is needed to schedule as many links as possible for Λ at slot t. This is reasonable because one objective in Eq. (3) aims to concurrently schedule most links in each channel layer and minimize the maximum turnaround time. We need to know how to combine maximum link patterns and multiple paths. Those nodes shared by several paths are more likely to run out of free interfaces or channels. For example, no matter what paths are selected for the two pairs of (H, I) and (G, E) in Fig. 1, they always intersect with the path of pair (A, D). Hence, some nodes on $p_{(A,D)}$ are very likely to be overloaded.

Link Patterns At time slot t, a node can be involved in a link if it has common free channels with its neighbors and free transceivers. If a node does not have other idle transceivers, it cannot establish a link to another neighbor, even if it has free channels. Likewise, if a node does not have any interference-free channel because of the current active links and neighbor links, it can not establish another link, even if it has idle transceivers.

WMPP is critical to satisfy specific traffic requests while promising MUN or achieving minimum time. It is essential to explore efficient algorithms for routing and scheduling multi-pair traffic requests.

The total number of compatible paths may be affected by mesh parameters: the topology, available orthogonal channels, mesh size, router type, node interfaces, and interference model. Additionally, it is also limited by the node properties such as power strength and antenna type. For simplicity, we focus our discussion on omnidirectional antennas in the mesh mode, i.e., a sender has only one desired receiver.

A link pattern collects all interference-free links over channel c_i in static wireless meshs [8]. First of all, it makes full use of channel resources. Specifically, given a finite number of orthogonal channels, paths whose lengths are close to the mesh diameter are limited even if some local regions have free resources. To use remaining resources, some shorter paths should be scheduled simultaneously. This necessitates detecting maximal link pattern for channel c_i of planar mesh as in Fig. 1.

With globally pre-computed link patterns, scheduling specific link patterns should be synchronised. After the system collects the information of router nodes, including position, interface number, and radio power, a series of link patterns can be generated using Algorithm 1.

Algorithm 1 facilitates the most links over each channel for $\{(s_i, d_i)\}, i = 1, 2, \cdots, k$. We use $\otimes$ to denote an operation for assigning a specific feasible channel by keeping the least channels used for a path. This operation collects the link patterns together, which are distinguished to each other according to time t and channel c. Obviously, the output of link patterns are dominated by path. The output saves the compatible link patterns for each channel layer. Even the traffic request list can be used to evaluate the heuristic start point for selecting interference links, the fairness should also be considered with maximal link patterns for each channel-layered mesh.

Algorithm 1 collects the interference-free links for desired paths over each channel with a heuristic start link. Different initial links result in different link patterns. Other factors that affect the size of a link pattern include topology and interference model. Furthermore, this algorithm can be modified to compute all possible link patterns without duplication for every heuristic start in a given mesh.

Example 1 *Link $E\overrightarrow{c_1}B$ can coexist with $H\overrightarrow{c_1}F$ or $F\overrightarrow{c_1}H$ in Fig. 3. However, as link $B\overrightarrow{c_2}G$ is located at the center of the pruned mesh, any other links over this channel will be in conflict with it. Hence, link $B\overrightarrow{c_2}G$ exclusively uses channel c_2. Similarly, link $G\overrightarrow{c_3}H$ can coexist with $E\overrightarrow{c_3}D$ or $E\overrightarrow{c_3}A$ or $D\overrightarrow{c_3}E$ or $A\overrightarrow{c_3}E$, at most two links in any*

Algorithm 1 Paths by Link-Patterns
Input: Γ
Output: $\mathcal{P}$

Require: $\{(s_i, d_i)\} \neq \emptyset$
Ensure: $\xi \neq 0$
$\mathcal{P} \Leftarrow \emptyset$;
$Sort(\Lambda)$;
while $i < \rho$ and $t < \mathcal{T}$ **do**
  if $\exists i = i_0$, such that $p_{(s_{i_0}, d_{i_0})}$ satisfies each node on the path has free radios and interference free channels at time slot t **then**
    for $h = 1$ to h_{i_0} **do**
      $L^{i_0,h}$ collects all interference-free links with $l^{i_0,h}$ of $p_{(s_{i_0}, d_{i_0})}$;
      $P_c^{i_0,h} \Leftarrow (\{l^{i_0,h}\} \cup L^{i_0,h}) \otimes c$;
      $P^i \Leftarrow \oplus P_c^{i_0,h}$;
    $i \Leftarrow i + 1$;
  else
    $t \Leftarrow t + 1$;
    Select $(s_{i_1}, d_{i_1}) \in \{(s_i, d_i)\} - \{(s_{i_0}, d_{i_0})\}$;
return $\mathcal{P} = \cup_i P^i$;

combination. Additionally, link $A\overrightarrow{c_4}B$, which shares B with the former decomposed path $p < E, H >$, can coexist with $H\overrightarrow{c_4}G$, or $F\overrightarrow{c_4}G$, or $H\overrightarrow{c_4}F$, or $F\overrightarrow{c_4}H$. Finally, link $B\overrightarrow{c_5}C$ can coexist with $H\overrightarrow{c_5}F$ or $F\overrightarrow{c_5}H$.

The largest number of activated paths is upper bounded by the largest size of those link patterns over all channels. A path is established if every component hop is realized over a certain channel. Given a WMN with the parameters to form a CPG, the number of active paths is obviously no more than the maximum number of all link patterns over all channel layers. Considering the available channels, CA and link cooperation, the number of activated paths may be far less, because it is impossible to have the same maximum number of link patterns for every hop of $p_{(s_i,d_i)}$.

For example, suppose that there are available channels $\{c_1, c_2, c_3\}$ at time t and the maximum link pattern contains 15 links. The maximum number of activated paths cannot surplus 15 even if every link is a certain hop of some path, even assuming that each link of the maximum interference-free link patterns can successfully get its entire path activated with enough over $\{c_1, c_2, c_3\}$.

4. Algorithm for Compatible Paths

To design an efficient algorithm for computing multiple pair paths to simultaneously transmit data packets for realtime services, like video conferences, we need to define several terms clearly.

In MIMO WMNs, a *link* is a transmission connection between a pair of neighbor nodes (sender and receiver) with a traffic request. The sender candidates S_C and receiver candidates R_C are essential to examine interference. The sufficient and necessary conditions for interference-free is discussed in our earlier work [11]. The channels can be viewed as a checking loop for picking up maximum links while preserving a continuous paths for a specific pair.

We consider triangular meshes and arbitrarily connected graphs. Let T be the time period to update parameters for repeatedly scheduling a certain sequential link pattern, containing slots t_i, $i = 1, 2, ..., T$. For node N_v, we use d_{N_v} to denote its free interface count, and c_{N_v} to denote its available channels. To assign a channel c to a set P of interference-free neighbor pairs, we define a function $c(P)$. If $c(P)$ assigns channel c_k to link group P, it is denoted as P_{c_k}. A path $p_{(s_i,d_i)}$ can transmit traffic λ_i if every one of its link can be active. To assign $l^{i,j}$ a channel, we use a function $c()$. $c(l^{i,j}) = l_c^{i,j}$ assigns a channel to the j-hop link of $p_{(s_i,d_i)}$. Link pattern is generated by recursively expanding partial solutions after interference screening. Algorithm 2 is a resource aware scheme to compute compatible paths based on the optimal model with CPG [10].

Algorithm 2 Compatible Paths
Input: Γ
Output: $\Re$

Ensure: Node clock synchronization in the WMN
for $v = 1$ to $|V|$ **do**
  update d_v;
  update c_v;
for $i = 1$ to ρ **do**
  $sort(\Lambda)$ in a decreasing order on key z_i;
  update $\{(s_i, d_i)\}$ sequence in the order of $sort(\Lambda)$;
$sort(\Omega)$ in a decreasing order of $\{\varpi_k\}, k = 1$ to ψ;
for $t = 0$ to T **do**
  while $i < \rho$ **do**
    $i = 0$;
    for $j = 0$ to h^i **do**
      if $\exists N_{v_1}, N_{v_2} \in p_{(s_i,d_i)}$ and
      (N_{v_1}, N_{v_2}) is the j^{th} hop and
      $(d_{N_{v_i}} > 0) \wedge (c_{N_{v_i}} > 0) \wedge (c_{N_{v_1}} \cap c_{N_{v_2}} \neq \emptyset), i \in \{1, 2\}$ **then**
        $P^{i,j} = P^{i,j} \cup \{(N_{v_1}, N_{v_2})\}$;
        Sort $\{(P^{i,j})\}$ in a non-decreasing order according to their sizes;
        $c(P^{i,j}) = P^{i,j}_{c_k}$;
        $k = k + 1$;
      else {at least one condition is not satisfied}
        $i = i + 1$;
  $t = t + 1$;
for all $t < \mathcal{T}$ **do**
  activate $c(P_j)$;
return $\Re = \bigsqcup_{k,t}^{i,j} \{P^{i,j}_{c_k}\}$ for Λ;

We use $c(p_{(s_i,d_i)})$ to denote a directed sequence of links that allow real-time streams. It can be expressed as node sequence combining channel information. For example, $p_{(A,C)}$ can be implemented as links $(A, B)_{c_1}, (B, C)_{c_3}$. Meanwhile, path $p_{(E,H)}$ can be implemented as $(E, B)_{c_5}, (B, G)_{c_7}, (G, H)_{c_2}$ in Fig. 1.

This algorithm attempts to find more active multiple pair paths with more interference-free links over independent orthogonal channels, which naturally leads to a higher level of network resource utilization. The observations of Couto *et al.* [14] provide another perspective to find more compatible links for those selected paths. In fact, they realized that "the shortest path is not sufficient" through two testbed-based experiments, as minimum-hop routing often chooses routes that have significantly less capacity than the best link quality paths.

Path selection does affect the WMN performance. In Fig. 1, suppose that the interface counts of router nodes $\{A, B, C, D, E, F, Q, X, Y, Z\}$ are $\{2, 4, 2, 2, 3, 2, 2, 2, 2, 2\}$, respectively. The multiple pairs are $(A, D), (C, F), (G, J), (G, H), (I, E)$, and the corresponding traffic request queues are $\{4, 3, 2, 2, 2\}$. If we simply select the shortest path, the three paths would be $A\overrightarrow{c_1}B\overrightarrow{c_2}D$, $C\overrightarrow{c_3}B\overrightarrow{c_4}J\overrightarrow{c_5}F$, and $G\overrightarrow{c_5}A\overrightarrow{c_6}J$. However, if we replace path $C\overrightarrow{c_3}B\overrightarrow{c_4}J\overrightarrow{c_5}F$ by $C\overrightarrow{c_3}D\overrightarrow{c_4}E\overrightarrow{c_5}F$, at least the following four paths can become active simultaneously: $A\overrightarrow{c_1}B\overrightarrow{c_2}D$, $C\overrightarrow{c_3}D\overrightarrow{c_4}E\overrightarrow{c_5}F$, $G\overrightarrow{c_5}B\overrightarrow{c_6}J$, and $X\overrightarrow{c_5}A\overrightarrow{c_6}Y$. Note that the number of compatible paths in the second one increases by one, and its link count increases as well, from step 7 to step 9. Therefore, the second scheme is a better choice.

We use V^c to denote the set of nodes that still have free channels in V, and V^r to denote the set of nodes that still have free radio interfaces in V. $d(s_i, r_i) = 1$ means that (s_i, r_i) is a neighbor pair. The remaining free channel set of s_i is denoted as s_i^c.

To find other potential links, the procedure $Game\_Supplement$ is used to exploit chances to maximize the utility of mesh radio and channel resources.

Algorithm 3 $Game\_Supplement$
Input: The updated Γ after Algorithm 2
Output: $\Re'$

```
Require: V^c ≠ ∅
Ensure: V^r ≠ ∅
ℜ' = ∅;
while ∃(s_i, r_i) ∈ V^c ∩ V^r ∧ d(s_i, r_i) = 1 do
    if λ_i > 0 then
        l_{c_{i0}}^{(s_i,r_i)}, where c_{i0} ∈ s_i^c ∩ r_i^c;
        if l_{c_{i0}}^{(s_i,r_i)} is interference-free to ℜ' then
            ℜ' = ℜ' ∪ {l_{c_{i0}}^{(s_i,r_i)}};
            Exit While;
        else {confliction with other simultaneous links}
            while s_i^c ∩ r_i^c ≠ ∅ do
                c_{i0} = c_{i0} + 1;
                (s_i^c ∩ r_i^c) = (s_i^c ∩ λ_i^c − {c_{i0}});
        update V^c;
        update V^r;
return ℜ';
```

In fact, the procedure $Game\_Supplement$ is to enlarge the scale of the scheduled links as much as possible. However, $Game\_Supplement$ aims to use those free resources by picking more links. There are chances to get a whole path by taking more hops than the shortest path of the pair. Suppose the nodes with idle transceiver form V^r after Algorithm 2, and the nodes keeping available channels form V^c.

Assigning a channel c_{i_0} to link λ_i in Algorithm 3 implies confliction avoidance to $\Re$. The paths of final solution, combining Algorithm 2 and 3 form scheduling models for MIMO WMNs. Of course, the Algorithm 3 does not always promise additional path contribution. It actually works when the channel and radio count increases.

Some other major factors on compatible paths are node interface count, available channel count, mesh topology (node relatively position), heuristic initial neighbor pair, and antenna type (omnidirectional antenna, directive antenna, smart antenna, etc.). In real applications, the environments also affect the actual paths [3, 21].

5. Performance

The performance of Algorithm 2 is evaluated both analytically and by simulation. After we estimate the time complexity, we make some simulations on throughput, delay time, as well as statistics of active pairs. While T^r is given, simulations help to understand the performance influenced by different topologies.

5.1. Time Complexity

As mentioned in section 2, MPP in WMN is too hard to solve in exact algorithms. Algorithm 2 attempts to collect as many links as possible for every channel in the mesh to combine wanted paths. Note that $|S_C|$ reduces quickly along with the process of picking out more links without conflicts over a channel.

Let the senders of selected interference free links form a node set $\mathcal{S}$, and receivers form a node set $\mathcal{R}$. All neighbors of $\mathcal{S}$ is denoted as $\mathcal{S}_N$, while All neighbors of $\mathcal{R}$ is denoted as $\mathcal{R}_N$. According to rules as Table 1, for next link to add into the current link pattern, the selection range is given by S_C and R_C.
The next link sender candidates are in (4):

$$S_C = S_C - S - R - \mathcal{S}_N - \mathcal{R}_N. \tag{4}$$

The corresponding receiver candidate set is in (5):

$$R_C = R_C - \mathcal{S} - \mathcal{R} - \mathcal{S}_N. \tag{5}$$

We denote the average node degree of a given mesh as D_{Ave}. For example, in a triangular mesh, $D_{Ave} = 6$, while in a grid, $D_{Ave} = 4$.

An approximate estimate to the size of $|S_C| = |S_C| - 2D_{Ave}$. At the initial step,

$$|S_C| \approx |V| - 2D_{Ave}. \tag{6}$$

Generally, the recursion equation for the size of S_C is as following:

$$|S_C| = |S_C - \mathcal{S} - \mathcal{R} - \mathcal{S}_N - \mathcal{R}_N| \approx |S_C| - 2D_{Ave}. \tag{7}$$

The recursion equation for size of R_C is as follows:

$$|R_C| = |R_C - \mathcal{S} \cup \mathcal{R} - \mathcal{S}_N| \approx |R_C| - D_{Ave} \times |\mathcal{S}|. \tag{8}$$

At the initial step, the selection for both sender and receiver does not need interference screening. i.e, sender set and receiver set are empty. So, R_C in (8) changes to:

$$|R_C| \approx |V| - D_{Ave} \times |\mathcal{S}|. \tag{9}$$

Any link must have one node in S_C and one node in R_C. In other words, any link is an element of $S_C \times R_C$. Then, compatible links for a channel are a subset of relation $S_C \times R_C$.

In Algorithm 1, for one channel, selecting next compatible link goes on until $(S_C = \emptyset) \vee (R_C = \emptyset)$.

As the link patterns are computed via the heuristic start of traffic requests of multiple pairs, sorting the traffic requests in decreasing order costs at worst $\mathrm{O}(|V| \log |V|)$, even if the graph is complete graph $K_{|V|}$.

Another time consumption task is to screen the interference after updating S_C and R_C. Note that generating S_C and R_C only takes $\mathrm{O}(|V|)$. As for adding a link, two nodes from S_C and R_C must match as neighbors, i.e., the shortest distance of them is 1.

Meanwhile, to select as many links as possible, a preferred next sender is one of 2-hop away from the already selected senders $\mathcal{S}_s$. Those 2-hop away in S_C will be checked for next compatible link in priority. This is to avoid space and spectrum taken up by scattered links, because the algorithm aims to get more links for a link pattern in a path-dominated heuristic way.

The interference checking needs to find a neighbor pair of $(sender, receiver)$, where a new sender is 2-hop away to one of $\mathcal{S}_s$. A new receiver must satisfy the conditions in Table 1, while these two nodes are adjacent by checking the adjacent matrix of the mesh graph. The distance can also be verified by searching the distance matrix. This checking takes time at most of $\mathrm{O}(|V|^2)$.

Note that the size of S_C or R_C becomes smaller quickly according to (4) and (5). Let T'_i represent the i-th updated size of S_C. Let the compatible link computing process finish after k times calling of its procedure, where k is given by:

$$\begin{array}{l} T'_k = T'_{k-1} - 2D_{Ave}, \\ T'_0 = |V|, \\ T'_k = 0. \end{array} \tag{10}$$

After expanding (10), we obtain k by approximation:

$$k \simeq \left\lfloor \frac{|V|}{2D_{Ave}} \right\rfloor. \tag{11}$$

Now, we come to the conclusion for the time complexity T' of Algorithm 2. Because the next link is always determined after checking the interference, and along with the desired paths of some node pairs with traffic request priority, the link patterns from different channels work together to concatenate those paths.

$$T' < \mathrm{O}(|V|^2) + k \cdot |S_C| \cdot |R_C| < \mathrm{O}(|V|^3). \tag{12}$$

Therefore, the time complexity of Algorithm 2 is $\mathrm{O}(|V|^3)$. In broadband backbone WMNs, nodes are equipped with multiple interfaces, which can be viewed as the upper limit of a node on simultaneous links at a time slot. Hence, D_{Ave} is determined by the number of node interfaces, which is different from the node degree in the topology.

5.2. The simulations

We run several sets of simulations to evaluate the performances of Algorithm 2 over two topologies in Figure 4(b) and Figure 4(a): 61-node triangular mesh and 61-node arbitrary mesh. The arbitrary one is generated by selecting random positions for indexed nodes. In the triangular mesh, there are 4-hop circles surrounding the center node. We use the triangular one with additional aims to facilitate the discussion and illustrate the methods. Meanwhile, the randomly generated one is used to evaluate the robustness or verify the consistency for practical generality.

Furthermore, to evaluate the performance of Algorithm 2, we conduct simulations over these two topologies with variations in the number of channels, the number of nodes, and traffic queue sizes.

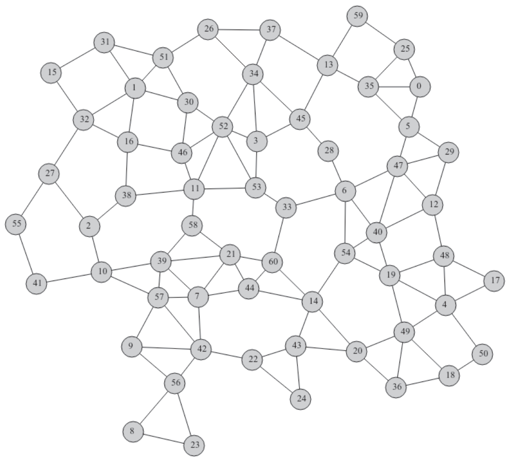
With different multiple pairs, where traffic queues are used to measure the sizes of traffic requests for those node pairs, we conduct simulations to estimate the maximum boundary for the combination cases of the numbers of interfaces and channels. To understand the throughput increase with variations in the available radios $\{4, 8, 12, 16, 20\}$ and the channels $\{8, 16, 32, 64, 128\}$, we run simulations in the situations of MPP for all pairs, MPP for some pairs with path crossing each other, and MPP with less resource competition.

These two network topologies are both virtually deployed in a 100×100 km^2 area with available radio number cases $\{4, 8, 12, 16, 20\}$, and channel number cases $\{8, 16, 32, 64, 128\}$. We use T_d to denote the effective transmission distance of certain power strength and I_d to denote the interference distance. We have $T_d < I_d$ and $I_d < 2T_d$. The interference scanning is under the conditions in Table 1. Time duration is set to be $5ms$, packet size is set to be $1MB$, and each link capacity is set to be $10Mb$. A period spans 0.5 second, equally, 100 time slots.

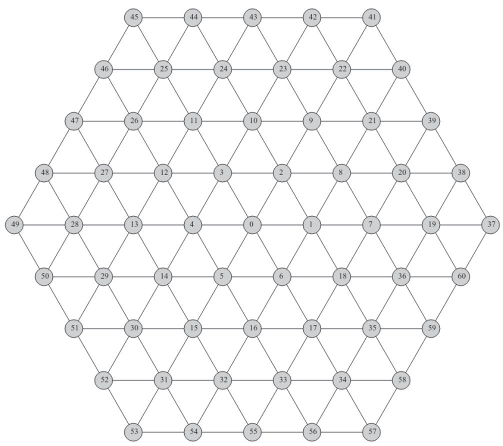
In a combinatorial way, the input instances vary with some parameters such as the number of router node interfaces, the number of available orthogonal channels, the specific multiple pairs and the corresponding traffic queue sizes. We design total 25 combinations of the channel number and the radio number over two topologies to evaluate the performance. The traffic matrix to all pairs is in form $(250)_{61\times61}$.

We first consider a traffic model to simulate the situation of a very busy backbone network, where each node has a traffic request to every others.The traffic matrix to all pairs is $(250)_{61\times61}$.

The throughput comparison over the two topologies is shown in Fig. 5. To be concise, in all figures, we use $(R = Radio_{num}) \wedge (C = Channel_{num})$ to represent a specific combination case, where nodes are equipped with R interfaces, and the mesh operates over C available orthogonal channels. We are able to draw conclusions on the proper relation between radio number and channels for both efficiency and economic purposes: a higher throughput improvement from case $(R = 12) \wedge (C = 128)$ to case $(R = 16) \wedge (C = 128)$ than that from case $(R = 16) \wedge (C = 128)$ to case $(R = 20) \wedge (C = 128)$. This observation is true for both topologies. Then, we conclude that the economically efficient case is $(R = 16) \wedge (C = 128)$. It is clear that the improvements are significant between channel variations for the case $R = 16$. Additionally, the throughput of triangular mesh outperforms that of the random one by $200MB/s$. Meanwhile, we observe that the performance is also stable in the arbitrary mesh, compared with that of the carefully planned triangular mesh.



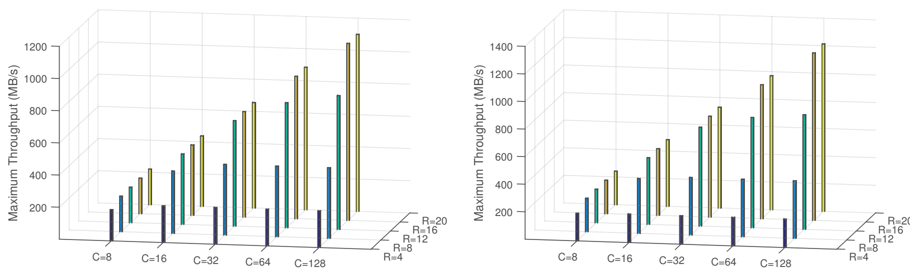
(a) A randomly generated mesh for indexed nodes



(b) A carefully planned triangular mesh

Fig. 4. The arbitrary mesh and the triangular mesh.

To evaluate the general efficiency of the algorithm, the average delays for 25 cases are shown over two topologies in Figure 6. Given a specific traffic request, the combined 25 cases are the elements of $R \times C$, i.e. $\{4, 8, 12, 16, 20\} \times \{8, 16, 32, 64, 128\}$. For example, $(4, 32)$ means that the number of interfaces is 4, and the number of available channels is 32. The average delays in these two topologies are simulated independently. The overall delays are smaller in triangular mesh than that in random mesh, which matches our theoretical expectation. If we only consider the best combined cases, the triangular mesh and the random mesh reveal the common fact: the cases of $R = 20 \wedge R = 16$ with $C = 64 \wedge C = 128$ are efficient, because $R = 16 \wedge C = 64$ or $R = 16 \wedge C = 128$ show less average delays than other cases.

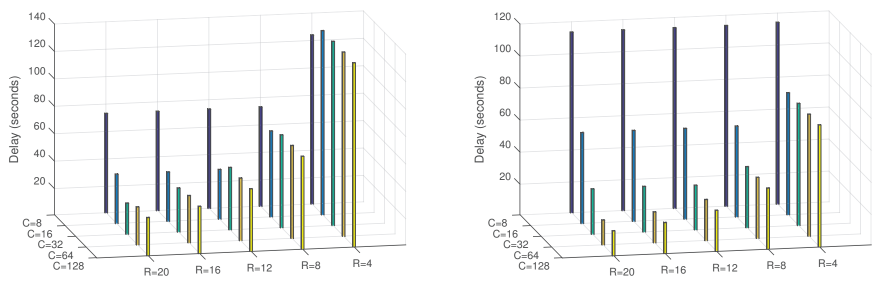


(a) Maximum throughput in the random mesh (b) Maximum throughput in the triangular mesh

Fig. 5. Maximum throughput of Algorithm 2 over two topologies.

The difference of the total used time between triangular mesh and random mesh is significant. As shown in Figure 6, triangular mesh has significant improvements in 25 cases. For example, in case $R = 4 \wedge C = 8$, triangular mesh uses almost only half time of random mesh for the same multiple pairs and traffic queues.

The simulation results also show that the proposed algorithm works efficiently in terms of delay, and it performs better in the triangular mesh than the arbitrary one, as shown in Fig. 6.



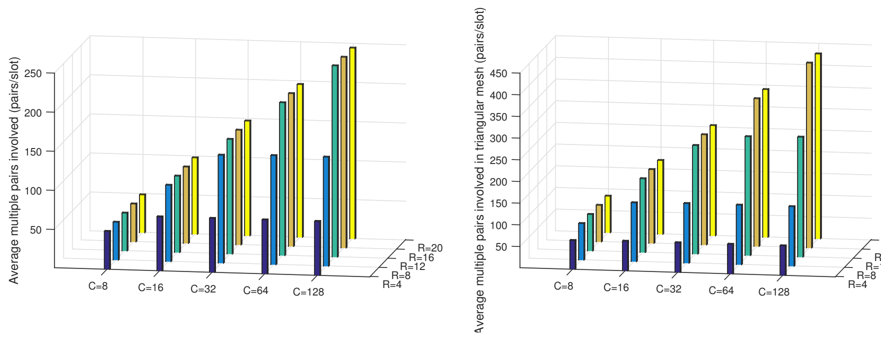
(a) The delay over the random mesh (b) The delay over the triangular mesh

Fig. 6. Average delays of Algorithm 2 over two topologies.

Algorithm 2 is evaluated by 22 random pairs, which are: $(N_{37}, N_0), (N_{41}, N_{43}), (N_{39}, N_{21}), (N_{38}, N_{33}), (N_{57}, N_6), (N_{18}, N_{52}), (N_{55}, N_{10}), (N_{29}, N_{20}), (N_3, N_{36}), (N_{22}, N_9), (N_{25}, N_{44}), (N_{43}, N_{42}), (N_{41}, N_{53}), (N_{39}, N_{38}), (N_{60}, N_{48}), (N_{34}, N_0), (N_{54}, N_0), (N_{45}, N_{57}), (N_{29}, N_0), (N_{16}, N_{20}), (N_{23}, N_{37}), (N_{11}, N_{29})$.

The specific traffic queue sizes of T^r are assigned as: $\alpha = \beta = (70, 60, 50, 40, 30, 20, 10, 70, 60, 50, 40, 30, 20, 10, 80, 50, 60, 70, 40, 30, 20, 30)$. The number of orthogonal channels $|\Omega|$ can be $\{4, 8, 12\}$, and the number d of node interfaces can be one of $\{3, 4, 6\}$. $d = 3$ represents a hexagonal mesh, $d = 4$ represents a grid mesh, and $d = 6$ represents a triangular mesh. Note that in an arbitrary mesh, a node degree is determined by randomly distributed node positions.

The average number of pairs involved We calculate the statistics on the number of pairs involved in the scheduling of each time slot to show the maximum, average and minimum pairs involved. The number of pairs is partially related to the network utility rate, and can be used to evaluate the topology efficiency at the topology planning stage as well. The average number of pairs per time slot is illustrated in Fig. 7, corresponding to the two 61-node topologies, the random and triangular mesh, respectively.



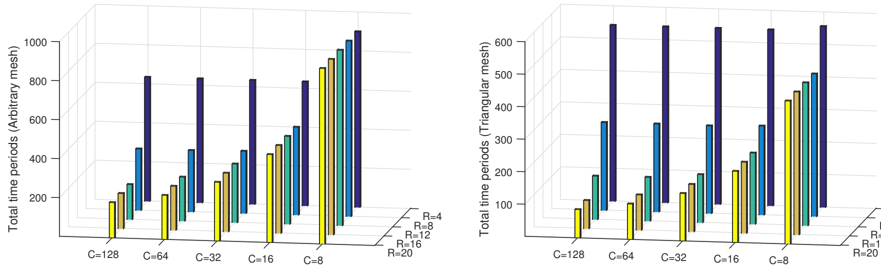
(a) The average number of pairs scheduled over the random mesh

(b) The average number of pairs scheduled over the triangular mesh

Fig. 7. The average number of pairs scheduled by Algorithm 2.

The time used to finish traffic queues The algorithm works on both topologies of triangular mesh and randomly generated mesh. Specifically, Fig. 8 illustrates the total time for traffic $(250)_{61\times61}$ over two topologies, where each node has a request of 250-packet traffic to every other one. We observe that the total time costs are consistent, for both of the triangular mesh and random mesh topologies. Here, one period is set to be 0.5 second, i.e., 100 time slots. Fig. 8(a) and 8(b) plot the total time used for all combined cases in the randomly generated mesh and in the triangular mesh, respectively. Fig. 8 also shows the effects of the number of interfaces and the number of channels. More node interfaces result in less time cost for the specific traffic size. More available channels also result in less time cost. The most efficient case is among $R = 16 \wedge C = 128$, $R = 20 \wedge C = 128$, $R = 16 \wedge C = 64$ and $R = 20 \wedge C = 64$.

We further make a comparison with AODV over the triangular and arbitrary meshes. AODV achieves almost less than half of the throughput achieved by Algorithm 2. A care-



(a) Time used over arbitrary mesh (b) Time used over triangular mesh

Fig. 8. Time used to transmit the traffic $(250)_{61\times 61}$.

ful investigation into AODV execution processes reveals that AODV does not take into account the remaining resources and the topology updates. For example, even if there is a resource-free shortest path for a given pair, AODV may select another longer path with less resources to forward packets. This random choice certainly degrades its performance in MRMC WMNs. Also, AODV leads to a lower performance in our combinatorial cases as it does not fully consider MRMC situations.

6. Conclusion

WMPP is raised from real applications for data, voice and video transmission in WMNs. With the CPG model and an in-depth analysis, we develop a joint routing and scheduling scheme through channel layered interference-free links, aiming to maximize compatible paths to provide the highest Quality of Service over limited resources. We proposed to decompose multiple paths into channel layered interference-free link patterns to maximize the resource use in MIMO WMNs. Since link patterns mainly contain links of the paths for multiple pairs, maximum compatible paths naturally result in the maximum utilization of network resources for a given problem instance. Extensive simulations over triangular and arbitrary topologies show that the proposed optimization scheme computes maximum link patterns efficiently and exhibits a stable performance, which meets our theoretical expectation. It is our further interest to conduct more extensive simulations for the deployment of BS nodes in triangular and arbitrary meshes.

Acknowledgment. The research is partially funded by China Scholarship Council (CSC) No. 2013-06755013, and is also partially sponsored by U.S. National Science Foundation under Grant No. CNS-1560698 with New Jersey Institute of Technology.

References

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

2. Alicherry, M., Bhatia, R., Li, L.E.: Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In: Proceedings of the 11th MobiCom. pp. 58–72. ACM (2005)
3. Andersen, D.G., Snoeren, A.C., Balakrishnan, H.: Best-path vs. multi-path overlay routing. In: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement. pp. 91–100 (2003)
4. Andrews, M., Zhang, L.: Hardness of the undirected congestion minimization problem. SIAM Journal on Computing 37(1), 112–131 (2007)
5. Andrews, M., Zhang, L.: Almost-tight hardness of directed congestion minimization. Journal of the ACM (JACM) 55(6) (2008), article ID 27
6. ASSOCIATION, I.S.: Ieee std. 802.16-2004 for local and metropolitan area networks, part 16: Air interface for fixed and mobile wireless access systems. http://standards.ieee.org/getieee802/download/802.16-2004.pdf (October 2004)
7. Cao, L., Zheng, H.: On the efficiency and complexity of distributed spectrum allocation. In: Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007. 2nd International Conference on. pp. 357–366 (2007)
8. Cao, Z., Peng, L.: Destination-oriented routing and maximum capacity scheduling algorithms in cayley graph model for wireless mesh network. Journal of Convergence Information Technology 5(10), 82–91 (2010)
9. Cao, Z., Tang, J.: Routing methods and scheduling patterns in mimo wmn virtual model. Applied Mechanics and Materials 519, 216–221 (2014)
10. Cao, Z., Wu, C.Q., Berry, M.L.: On routing of multiple concurrent user requests in multi-radio multi-channel wireless mesh networks. In: 17th International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2016, Guangzhou, China, December 16-18, 2016. pp. 24–29 (2016)
11. Cao, Z., Wu, Q., Zhang, Y., Shiva, S.G., Gu, Y.: On modeling and analysis of mimo wireless mesh networks with triangular overlay topology. Mathematical Problems in Engineering (2015), article ID 185262
12. Cao, Z., Xiao, W., Peng, L.: A mesh$\times$ chain graph model for mimo scheduling in ieee802. 16 wmn. In: Proc. of the 2nd IEEE International Conference on Computer Modeling and Simulation. vol. 2, pp. 547–551 (2010)
13. Cappanera, P., Lenzini, L., Lori, A., Steay, G., Vaglini, G.: Optimal link scheduling for real-time traffic in wireless mesh networks in both per-flow and per-path frameworks. In: World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a. pp. 1–9. IEEE (2010)
14. De Couto, D.S., Aguayo, D., Chambers, B.A., Morris, R.: Performance of multihop wireless networks: Shortest path is not enough. ACM SIGCOMM Computer Communication Review 33(1), 83–88 (2003)
15. Giannoulis, A., Salonidis, T., Knightly, E.: Congestion control and channel assignment in multi-radio wireless mesh networks. In: Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on. pp. 350–358. IEEE (2008)
16. Godsil, C., Royle, G.: Algebraic graph theory, vol. 207. Springer (2001)
17. Karp, R.M.: Reducibility among combinatorial problems. Springer (1972)
18. Laneman, J.N., Tse, D.N., Wornell, G.W.: Cooperative diversity in wireless networks: Efficient protocols and outage behavior. IEEE Transactions on Information Theory 50(12), 3062–3080 (2004)
19. Larsson, E.G., Edfors, O., Tufvesson, F., Marzetta, T.L.: Massive mimo for next generation wireless systems. IEEE Communications Magazine 52(2), 186–195 (2014)
20. McDiarmid, C., Reed, B.: Channel assignment and weighted coloring. Networks 36(2), 114–117 (2000)

21. Nandiraju, N., Nandiraju, D., Agrawal, D.: Multipath routing in wireless mesh networks. In: IEEE international conference on Mobile adhoc and sensor systems (MASS). pp. 741–746. IEEE (2006)
22. Ramachandran, K.N., Belding-Royer, E.M., Almeroth, K.C., Buddhikot, M.M.: Interference-aware channel assignment in multi-radio wireless mesh networks. In: INFOCOM. vol. 6, pp. 1–12 (2006)
23. Ramanathan, S., Lloyd, E.L.: Scheduling algorithms for multihop radio networks. IEEE/ACM Transactions on Networking (TON) 1(2), 166–177 (1993)
24. Raniwala, A., Gopalan, K., Chiueh, T.c.: Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. ACM SIGMOBILE Mobile Computing and Communications Review 8(2), 50–65 (2004)
25. Schumacher, A., Haanpää, H.: Distributed network utility maximization in wireless networks with a bounded number of paths. In: Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks. pp. 96–103. ACM (2008)
26. Sen, A., Huson, M.L.: A new model for scheduling packet radio networks. Wireless Networks 3(1), 71–81 (March 1997)
27. Suurballe, J.W., Tarjan, R.E.: A quick method for finding shortest pairs of disjoint paths. Networks 14(2), 325–336 (1984)
28. Wang, I.L., Johnson, E.L., Sokol, J.S.: A multiple pairs shortest path algorithm. Transportation science 39(4), 465–476 (2005)

Zhanmao Cao received his PhD from School of Computer Science and Technology, South China University of Technology, Guangzhou, China. He is Associate Professor, Dept of Computer Science, South China Normal University. He is interested in developing algorithms for application problems. He proposed algorithms BTA and DC-BTA in multiple sequence alignment of bioinformatics. He is now working on MIMO WMN model, routing and scheduling algorithms.

Chase Qishi Wu received his PhD in Computer Science, Louisiana State University (LSU), Baton Rouge, LA. He is tenured Associate Professor, Department of Computer Science, New Jersey Institute of Technology. He is very active in both research projects and published papers. His research covers a few areas: Big data, data-intensive computing, parallel and distributed computing, high-performance networking, large-scale scientific visualization, wireless sensor networks, cyber security.

Mark L. Berry is a PhD student in Wu's group, Department of Computer Science, New Jersey Institute of Technology.

Received: January 24, 2017; Accepted: July 29, 2017.