A high reliability and low latency routing algorithm in
cognitive wireless mesh networks

Zhufang Kuang*

School of Computer and Information Engineering, Central South University of Forestry and Technology, Changsha 410004, China and School of Software, Central South University, Changsha 410083, China Email: zfkuangcn@csu.edu.cn *Corresponding author

Zhigang Chen
School of Software, Central South University, Changsha 410083, China Email: czg@csu.edu.cn

Abstract: A channel utility value computing (CUVC) algorithm is proposed. The channel utility value contains channel usage probability and channel stability. A high reliability and low latency wireless link weight computing algorithm (RL2W) is proposed. On this basis, a high reliability and low latency routing and spectrum allocation algorithm based on dynamic programming in cognitive wireless mesh networks (HRL2A) is proposed. High reliability and low latency route is the objective of HRL2A. Firstly, HRL2A computes the channel utility value using CUVC for constructing the high reliability route. Secondly, HRL2A uses the algorithm RL2W computing wireless link weight. Thirdly, the route of high reliability and low latency is constructed based on dynamic programming, and the wireless link channel is allocated. Simulation results show that HRL2A algorithm can achieve expectation goal. The construction route not only has higher reliability, but also has lower latency. The throughput has been increased.

Keywords: cognitive wireless mesh networks; CWMNs; routing; spectrum allocation; high reliability; low latency.


Biographical notes: Zhufang Kuang received his PhD in Computer Science from Central South University, Changsha, Hunan, China, in 2012. He is a post-doctoral of School of Software, Central South University, and an Associate Professor of School of Computer and Information Engineering, Central South University of Forestry and Technology. He is a senior member of
A high reliability and low latency routing algorithm in CWMNs

CCF, and a member of IEEE and ACM. His research interests include next generation broadband wireless communication systems, cognitive wireless mesh networks.

Zhigang Chen received his BSc, MSc and PhD degrees from Central South University, China, 1984, 1987 and 1998, respectively. He is a Full Professor of School of Software, Central South University, China. He is a senior member of CCF, a member of IEEE and ACM. His research interests include network computing and distributed processing.

1 Introduction

Cognitive radio (Mitola and Maguire, 1999) (CR) is intelligent revolutionary spectrum (channel) sharing technology and the most important new wireless technology today. The core function of CR is that it can sense the vacancy spectrum resources and share these unused spectrum resources. Secondary users (SU) can use the authorised spectrum which primary users (PU) did not use (Akyildiz et al., 2006).

A cognitive wireless mesh network (CWMN) is a combination of wireless mesh network (Chen et al., 2007; Wu et al., 2010) which integrates CR technology. A CR-Mesh node (such as a CR-Mesh gateway, a CR-Mesh router, or a CR-Mesh client), which integrates CR technology, can sense the spectrum which PUs are not using and access the vacancy spectrum resources (Bouabdallah et al., 2011; Jia et al., 2013). The spectrum sensing algorithm does not research in this paper.

The problem of routing and spectrum allocation in CWMN is studied in this paper. High reliability and low latency route is the objective.

The routing protocol in wireless mesh network cannot be directly used in CWMN (Cesana et al., 2011). At present, about the routing problem in CWMNs, there are already some research results (Al-Rawi and Yau, 2013; Dong et al., 2010; Amini and Dziong, 2014; El-Sherif and Mohamed, 2014; Soltani and Mutka, 2013; Yuan et al., 2013; Ding and Xiao, 2013; Li, 2013; Mourey et al., 2012; Zhang et al., 2010; Kuang et al., 2011; Misra et al., 2014; Capdehourat et al., 2016; Venkatesan and Vijayarangan, 2016; Jia et al., 2015; Soltani and Mutka, 2013; Hou et al., 2015; Moustafa et al., 2014).

Aim at solving the problem of coexistence of CWMN and other wireless networks, in order to share spectrum among multiple wireless networks, a routing and spectrum allocation algorithm with the objective of minimise the number of used channels in CWMN was proposed by Dong et al. (2010). A problem solving framework based on economic with the final goal of the network profit maximisation was proposed by Amini and Dziong (2014). Aim to minimise the end-to-end delay for joint optimisation of routing and resource allocation problem, the problem is formulated as a nonlinear integer programming problem, and then Lagrange dual method was proposed based on a distributed solution in CWMN by El-Sherif and Mohamed (2014).

The routing problem in video stream of multimedia applications in CWMN is studied by Soltani and Mutka (2013). The problem is transformed into a decision theory problem firstly, and then a video stream aware cognitive routing strategy VCR was proposed. With the optimisation of minimising the sum interference of all CR-Mesh nodes to PU, a distributed routing algorithm based on network game in CWMN was proposed by
Yuan et al. (2013). A multisource video on-demand application over a CWMNs was studied with the objective of maximising the number of sessions of the network. A distributed multipath routing and spectrum allocation algorithm (DRCA) was proposed by Ding and Xiao (2013). The total bandwidth consumption of each session is small, while the number of simultaneous sessions more.

A geographic routing protocol GCM for large-scale heterogeneous mixture CWMN was proposed by Li (2013). GCM is a multi-objective routing protocol with energy, spectrum sensing load and link quality, etc. With the optimisation of maximising end-to-end throughput, a routing and channel assignment algorithm based on dynamic programming in CWMN was proposed by Mumey et al. (2012). Zhang et al. (2010) proposed distributed channel allocation strategy for a joint routing. The main purpose of this channel allocation strategy is to maintain the channel diversity in the field. The optimisation performance index is the average throughput and the average delay. With the objective of maximising wireless business acceptance rate, a centralised multi path routing and spectrum allocation algorithm SA2JR was proposed in literature (Kuang et al., 2011), which has large communication overhead. A CR-based dynamic bandwidth allocation scheme for SUs in a cluster-based WiMAX network is proposed by Misra et al. (2014). Which uses a learning automata-based algorithm to find the optimal transmission channel, while ensuring minimum channel loss and a considerably high SIRN, and concurrently minimising costly channel switching activities when PUs request licensed channels. An optimum spectrum allocation mechanisms for a cognitive wireless multihop mesh network is proposed by Capdehourat et al. (2016). Which introduce a stochastic model to formulate the problem, considering PUs’ activity and a periodically scheduled assignment scheme. Venkatesan et al propose a secure and reliable routing in CRN based on distributed Boltzmann-Gibbs learning algorithm (Venkatesan and Vijayarangan, 2016). The throughput maximisation problem jointly with power control, channel allocation and routing under SINR model is researched by Jia et al. (2015). Soltani and Mutka (2013) use a decision theory framework to model the problem of routing under uncertainties involved in a CR network. A utility function is designed to capture the effect of spectrum measurement, fluctuation of bandwidth availability, and path quality. A spectrum- and energy-efficient routing (SEER) protocol for CRAHNs is proposed by Hou et al. (2015), which improves transmission efficiency and balances energy consumption. The design of the protocol uses the techniques of expected transmission power, expected residual lifetime and utility indifference curves.

With the objective of high reliability and low delay, the problem of routing and spectrum allocation in CWMNs was studied in this paper. This paper offers the following innovations when compared to existing research.

1. The problem of routing and spectrum assignment with the objective of high reliability and low end-to-end delay was studied in this paper.
2. The difference between the available channels, such as channel using probability, channel stability, channel delay, was considered.
3. A channel utility value computing (CUVC) algorithm was proposed, which includes channel using channel utility probability and channel stability. A link weight calculation algorithm RL2W was proposed, which includes reliability and delay. On the basis of this, a distributed routing and spectrum allocation algorithm HRL2A with the objective of high reliability and low end-to-end delay based on dynamic programming was proposed.
The remainder of the paper is organised as follows. We discuss the network model and problem description in Section 2. In Section 3 and Section 4, we describe the proposed CUVC algorithm and link weight calculation algorithm RL2W. Section 5 presents the HRL2A algorithm with the objective of high reliability and low end-to-end delay based on dynamic programming. Simulations comparing the performance of the proposed algorithms are presented in Section 6. Section 7 concludes the paper and outlines our future work.

2 Network model and problem description

2.1 Network model

We adopt a simple undirected graph $G = (V, E)$ model of the CWMN, which consists of CR-Mesh router and CR-Mesh gateways. $V$ represents the set of CR-Mesh routers, CR-Mesh gateways and CR-Mesh clients. $VG$ represents the set of CR-Mesh gateways. $VR$ represents the set of CR-Mesh routers. $VC$ represents the set of CR-Mesh clients. $E$ represents the set of wireless links. The physics distance between node $v_i$ and node $v_j$ is represented by $d(v_i, v_j)$.

Each node $v_i \in V$ has an available channel set $K_i$ which has been sensed. $Dy^k$ represents the delay of the channel $k (k \in K)$, in units of ms. Different channels have different delays (Cesana et al., 2011; Al-Rawi and Yau, 2013; Kuang et al., 2011), i.e., different channels $i$ and $j$, leads to $Dy^i \neq Dy^j$. Each node $v_i \in V$ has $I_i$ cognitive radio interfaces (CRIs).

$TR$ and $IR$ represent the communications distance and interference distance respectively, and $IR = 2 \times TR$.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V$</td>
<td>Set of CR-Mesh nodes</td>
</tr>
<tr>
<td>$E$</td>
<td>Set of edges</td>
</tr>
<tr>
<td>$K$</td>
<td>Set of available channels</td>
</tr>
<tr>
<td>$TR$</td>
<td>Communication distance</td>
</tr>
<tr>
<td>$IR$</td>
<td>Interference distance</td>
</tr>
<tr>
<td>$I_u$</td>
<td>Number of CRIs of node $u$</td>
</tr>
<tr>
<td>$K_u$</td>
<td>Set of available channels of node $u$</td>
</tr>
<tr>
<td>$Dy^k$</td>
<td>Delay of the channel $k$</td>
</tr>
<tr>
<td>$x(u, v)$</td>
<td>Allocated channel of link $(u, v)$</td>
</tr>
<tr>
<td>$K(u, v)$</td>
<td>Set of available channels of link $(u, v)$</td>
</tr>
<tr>
<td>$N(u)$</td>
<td>Set of neighbour nodes of node $u$</td>
</tr>
<tr>
<td>$N'(u)$</td>
<td>Set of adjacent nodes of node $u$</td>
</tr>
<tr>
<td>$P^k(u)$</td>
<td>The channel using probability value</td>
</tr>
<tr>
<td>$S^k(u)$</td>
<td>The channel stability value</td>
</tr>
<tr>
<td>$Pr^k(u, v)$</td>
<td>The using probability value in channel $k$ between node $u$ and node $v$</td>
</tr>
<tr>
<td>$Sh^k(u, v)$</td>
<td>The channel stability value in channel $k$ between node $u$ and node $v$</td>
</tr>
<tr>
<td>$f(u, v)$</td>
<td>Link weight of $(u, v)$ allocating channel $k$</td>
</tr>
<tr>
<td>$dp^*(u, v, k)$</td>
<td>The minimum path weight from node $u$ to destination node $v$</td>
</tr>
</tbody>
</table>
Two CR-Mesh nodes which can communicate with each other must satisfy the following conditions:

1. there are common available channels, \( K_i \cap K_j \neq \emptyset \)
2. there are unoccupied CRIs for each node
3. the nodes must satisfy the restriction of distance, \( d(v_i, v_j) < T_R \)
4. the nodes must satisfy the restriction of interference.

\( x(u, v) = k \) represents the wireless link \((u, v)\) is allocated channel \(k\). \( x(u, v) = 0 \) represents the wireless link \((u, v)\) is not allocated any channel. Every wireless link either is allocated only one channel, or is not allocated a channel. All nodes work in half duplex mode, and a common control channel (CCC) is used between each CR-Mesh node to transmit control information.

### 2.2 Problem description

We study the problem is under the condition of heterogeneous available channels, the route from source node to destination node is constructed distributedly. We aim to maximise the route reliability and minimise the average end-to-end delay. Wireless requests from the CR-Mesh gateway node to the CR-Mesh client nodes, namely \( \gamma_p = (g_p, c_p) \), \( g_p \) denotes the CR-Mesh gateway node, is source node of \( \gamma_p \), \( c_p \) denotes the CR-Mesh client node, is destination node of \( \gamma_p \). The route from \( g_p \) to \( c_p \) with the objective of high reliability and low end-to-end delay is constructed. Formal description can be written as formula (1)

\[
HL(\gamma_p) = \min \sum_{(u,v) \in \gamma_p} f^k(u,v) \left( p(g_p, c_p) \in PSet(\gamma_p) \right)
\]

\( HL(\gamma_p) \) represents the value of the reliability and delay of route from \( g_p \) to \( c_p \). \( p(g_p, c_p) \) represents a route from \( g_p \) to \( c_p \). \( PSet(\gamma_p) \) represents the route set of wireless request \( \gamma_p \). \( f^k(u,v) \) represents the wireless link weight value for link \((u, v)\), when it allocated channel \(k\). The calculation formula of \( f^k(u,v) \) include reliability and delay for link \((u, v)\). The calculation method described in Section 4.

### 3 Computing channel utility value

#### 3.1 Map fundamental

The problem of routing and spectrum assignment with the objective of high reliability and low end-to-end delay was studied in this paper. In the CWMMNs, due to the differences in CR-Mesh node location and the authorisation channel differences of PU, the sensing channel of different CR-Mesh node is heterogeneous. Therefore, this section first presents the basic idea of CUVC algorithm, and then shows the pseudo code description of CUVC algorithm.

CUVC algorithm for computing the channel utility value mainly includes: computing the available channel set between the CR-Mesh node and its neighbour nodes, and the using probability and stability for each available channel.
A high reliability and low latency routing algorithm in CWMNs

\[ K(v_i, v_j) \] represents the set of common available channels of node \( v_i \) and node \( v_j \), as shown in (2):

\[ K(v_i, v_j) = \{ k \mid \forall k \in K_i \cap K_j \} \] (2)

**Definition 1:** Adjacent node of \( v_i \): a node is within node \( v_i \) communication range, and the adjacent node set \( N'(v_i) \) is formulated as (3):

\[ N'(v_i) = \{ v_j \mid v_j \in V \&\& d(v_i, v_j) < T_R \} \] (3)

\( d(v_i, v_j) < T_R \) represents that the physical distance between the node \( v_i \) and node \( v_j \) is less than the distance of communication.

**Definition 2:** Neighbour node of \( v_i \): refers to the adjacent node of \( v_i \), the nodes have the common available channel with node \( v_i \), and the neighbour node set \( N(v_i) \) is formulated as (4):

\[ N(v_i) = \{ v_j \mid v_j \in N'(v_i) \&\& K(v_i, v_j) \neq \emptyset \} \] (4)

\( N'(v_i) \) represents the adjacent node set of node \( v_i \). \( K(v_i, v_j) \neq \emptyset \) represents that there are one common available channel between the node \( v_i \) and node \( v_j \) at least.

Every CR-Mesh nodes compute the using characteristic value for its’ all available channels, which includes the channel using probability value \( P^k(v_i) \) and the channel stability value \( S^k(v_i) \), which is given by formula(5) and formula(6)

\[ P^k(v_i) = \frac{\sum_{j=1}^{w} \alpha^k(t_j)}{w} \quad \forall k \in K_i, \quad \forall v_i \in V \] (5)

\[ S^k(v_i) = \frac{w - \beta^k(t_w)}{w} \quad \forall k \in K_i, \quad \forall v_i \in V \] (6)

In formula (5), \( P^k(v_i) \) represents the channel using probability value sensed by node \( v_i \) in channel \( k \). Node \( v_i \) counts whether or not channel \( k \) is used by PU from \( t_1 = 1 \). The detection time period is \( T \). \( \alpha^k(t_j) = 1 \) represents the channel \( k \) is used by PU. \( \alpha^k(t_j) = 0 \) represents the channel \( k \) is not used by PU. \( w \) represents the sum detection time period of node \( v_i \) up to present.

\[ \sum_{j=1}^{w} \alpha^k(t_j) \] represents the number of the channel \( k \) used by PU in a given time periods \( w \).

In formula (6), \( S^k(v_i) \) represents the channel stability value sensed by node \( v_i \) in channel \( k \), which represents the ratio from the ‘idle’ state into ‘using’ state in the total testing time cycle. \( \varepsilon^k(t_j, t_{j+1}) \) represents that whether or not \( \alpha^k(t_j) = 0 \) change into \( \alpha^k(t_j) = 1 \) from time \( t_j \) to \( t_{j+1} \) in the ‘idle’ into ‘using’. \( \varepsilon^k(t_j, t_{j+1}) = 1 \) represents the state of channel \( k \) from the ‘idle’ into ‘using’. Otherwise, \( \varepsilon^k(t_j, t_{j+1}) = 0 \).

\( \beta^k(t_w) \) represents the sum number of the state of channel \( k \) from the ‘idle’ into ‘using’ in a given time periods \( t_w \), which is given by formula (7)

\[ \beta^k(t_w) = \sum_{j=0}^{w-1} \varepsilon^k(t_j, t_{j+1}) \] (7)
Every node $v_i$ sends the available channel set $K_i$ and the channel using probability value $P_k(v_i)$ and the channel stability value $S_k(v_i)$ to its neighbouring nodes set $N'(v_i)$ through common control channel, namely, every node $v_i$ on every channel $k$, $\forall k \in K_i$, sends three tuple $Sat^k$ through CCC. $Sat^k$ is given by formula (8)

$$Sat^k = \{k, P_k(v_i), S_k(v_i)\}$$

Every node $v_i$ computes $K(v_i, v_j)$ with each node $v_j$, which is in $N'(v_i)$, and computes the channel using probability value $P_{nk}(v_i, v_j)$ and the channel stability value $S_{nk}(v_i, v_j)$ for channel $k$.

$P_{nk}(v_i, v_j)$ represents the using probability value in channel $k$ between node $v_i$ and node $v_j$, namely, the maximum value of $P_k(v_i)$ and $P_k(v_j)$ between node $v_i$ and node $v_j$, which is given by formula (9)

$$P_{nk}(v_i, v_j) = \max \left\{ P_k(v_i), P_k(v_j) \right\}$$

$S_{nk}(v_i, v_j)$ represents the channel stability value in channel $k$ between node $v_i$ and node $v_j$, namely, the minimum value of $S_k(v_i)$ and $S_k(v_j)$ between node $v_i$ and node $v_j$, which is given by formula (10)

$$S_{nk}(v_i, v_j) = \min \left\{ S_k(v_i), S_k(v_j) \right\}$$

### 3.2 CUVC algorithm

The pseudo code of CUVC algorithm is described as shown in Figure 1.

**Figure 1** The pseudo code description of the algorithm CUVC

```
Algorithm 1 CUVC algorithm
Input: $G = (V, E), v_i, v_j, K_i, K_j$
Output: $K(v_i, v_j), P_{nk}(v_i, v_j), S_{nk}(v_i, v_j), N(v_i)$
1) while ($\forall v_j \in N'(v_i)$){
2) computing $K(v_i, v_j)$ according to (2);
3) computing $N(v_i)$ according to (4);
4) } //end while
5) while ($\forall k \in K_i$){
6) computing $P_k(v_i)$ according to (5);
7) computing $S_k(v_i)$ according to (6);
8) } //end while
9) while ($\forall k \in K_j$){
10) send $Sat^k = (k, P^k(v_i), S^k(v_i))$;
11) } //end while
12) while ($\forall k \in K(v_i, v_j)$){
13) computing $P_{nk}(v_i, v_j)$ according to (9);
14) computing $S_{nk}(v_i, v_j)$ according to (10);
15) } //end while
```
4 Link weights computing algorithm

A wireless link weight computing algorithm RL2W was proposed in this section. The route reliability and end-to-end delay were included in RL2W. Firstly, RL2W algorithm computes the level from the CR-Mesh gateway node to all CR-Mesh routers and CR-Mesh node. Secondly, the set of parent nodes and children nodes of each node are computed. Finally, the weights of all wireless links are computed.

4.1 Computing level

The level from the CR-Mesh gateway node to all CR-Mesh router node and CR-Mesh client node is computed in this section. The level is namely the hop count from the CR-Mesh gateway node to all CR-Mesh routers and CR-Mesh client nodes in this paper. As for different CR-Mesh gateway nodes, the level of CR-Mesh router and CR-Mesh client node is different.

\[ L(g_i, v_j) \] represents the level from the CR-Mesh gateway node \( g_i \) to the CR-Mesh node \( v_j \).

The level computing algorithm RL2W is proposed in this section. Firstly, as for each CR-Mesh gateway node, the level from CR-Mesh gateway node to itself is set to 0, which is given by formula (11)

\[ L(g_i, g_i) = 0 \quad \forall g_i \in VG \quad (11) \]

Secondly, the BCM\((g_i, level)\) message is broadcasted through common control channel. The \( level \) is set to 0, namely, \( level = 0 \), which represents that the level from \( g_i \) to itself is 0.

As for each CR-Mesh router and client node, the level from CR-Mesh gateway node to it is set to \( \infty \), which is given by formula (12)

\[ L(g_i, v_j) = \infty \quad \forall g_i \in VG \quad \forall v_j \in V - VG \quad (12) \]

When the CR-Mesh router and client nodes receive the BCM\((g_i, level)\) message. As for CR-Mesh node \( v_j \) (\( \forall v_j \in V - VG \)), comparing \( L(g_i, v_j) \) with \( level + 1 \). If \( L(g_i, v_j) > level + 1 \), \( L(g_i, v_j) = level + 1 \), otherwise, \( L(g_i, v_j) \) does not change.

And then, as for CR-Mesh router node \( v_j \) (\( \forall v_j \in VR \)), the BCM\((g_i, level)\) message is broadcasted by node \( v_j \). The \( level \) in the message is set to \( L(g_i, v_j) \), namely, \( level = L(g_i, v_j) \). As for CR-Mesh client node \( v_j \) (\( \forall v_j \in VC \)), it does not broadcast the BCM\((g_i, level)\) message.

The pseudo code of LCA algorithm is described as show in Figure 2.
Figure 2  The pseudo code description of the algorithm LCA

Algorithm 2 LCA algorithm

Input: \( G = (V, E), u, N(v) \)
Output: \( L(g, v) \)

1) \( \text{if} (u \in V G) \{ // u = g_i \)
2) \( L(u, u) = 0; \)
3)  
4)  
5) \( \text{while}(\forall g_i \in V G) \{
6) \quad L(g_i, u) = \infty;
7) \quad // \text{end while}
8) \text{if} ( \text{receive} BCM(g_i, u, level) )\{
9) \quad \text{if} ( L(g_i, v) > \text{level} + 1 )\{
10) \quad \quad L(g_i, v) = \text{level} + 1;
11) \quad \} \text{else} \{
12) \quad \quad \text{Don not modify} L(g_i, v);\)
13) \quad \} \text{end if}
14) \text{if} (u \in V R) \{
15) \quad \text{Broadcast} BCM(u, level);\)
16) \} \text{end if}
17) \} \text{end if}
18) \} \text{end if}

4.2 Computing parent and children nodes set

After the level from the CR-Mesh gateway node to all CR-Mesh router node and CR-Mesh client node is computed. The set of parent nodes and children nodes of each node are computed in section. As for different CR-Mesh gateway nodes, the set of parent nodes and children nodes of each node is different.

As for CR-Mesh node \( u(\forall u \in V) \), the \( BLM(g_i, u, level) \) message is sent through common control channel to each neighbour node of \( N(u) \). The number of \( BLM(g_i, u, level) \) message sent by each node is \( |V G| \).

\( BLM(g_i, u, level) \) represents that the level from the CR-Mesh gateway node \( g_i \) to the CR-Mesh node \( u \) is set to \( level \), namely, \( level = L(g_i, v_j) \).

For all CR-Mesh nodes \( \forall u \in V \), upon receiving a \( BLM(g_i, u, level) \) message, comparing \( L(g_i, v) \) with the \( level \) value in the \( BLM(g_i, u, level) \).

If \( L(g_i, v) = level - 1 \), \( pre(g_i, v) = pre(g_i, v) \cup \{ u \} \).

\( pre(g_i, v) \) represents parent nodes set of node \( v \) relative to the gateway node \( g_i \).

If \( L(g_i, v) = level + 1 \), \( chi(g_i, v) = chi(g_i, v) \cup \{ u \} \).

\( chi(g_i, v) \) represents children nodes set of node \( v \) relative to the gateway node \( g_i \).
4.3 Computing link weight

Due to the differences in CR-Mesh node location and the authorisation channel differences of PU, the sensing channel of different CR-Mesh node is heterogeneous. The wireless link weight is computed in this section, when the link is pre allocated the channel \(k\).

As for CR-Mesh node \(v_j (\forall v_j \in V)\), the link weight \(f^k(v_j, v_q)\) for link \((v_j, v_q)\) allocating channel \(k\) is computed. \(v_q\) represents the children node of node \(v_j\) relative to the gateway node \(g_i \in VG\), namely, \(v_q \in chi(g_i, v_j)\). The formula \(f^k(v_j, v_q)\) is given by (13)

\[
f^k(v_j, v_q) = \frac{Pn^k(v_j, v_q) \times Dv^k}{\max\{Dv^k | \forall k \in K\} \times \max\{Dy^k | \forall k \in K\}}
\]

\(Pn^k(v_j, v_q)\) and \(Sn^k(v_j, v_q)\) represent the channel using probability and channel stability between node \(v_j\) and node \(v_q\) in common available channel \(k\) separately. \(Dv^k\) represents the delay in channel \(k\). \(\max\{Dv^k | \forall k \in K\}\) represents the maximum delay for all channels. \(Dv^k\) divided by \(\max\{Dv^k | \forall k \in K\}\) is designed to be delay normalisation.

The objective of high route reliability and low end-to-end delay is included in the link weight function \(f^k(v_j, v_q)\), \(Pn^k(v_j, v_q)\) represents the route reliability. \(\frac{Dv^k}{\max\{Dv^k | \forall k \in K\}}\) represents the delay. Because we hope that \(f^k(v_j, v_q)\) value the smaller the better, so the channel stability is in the denominator in \(f^k(v_j, v_q)\).

In order to construct the routing path with high reliability and low delay, the each hop of path need to have high reliability and low delay.

The pseudo code of RL2W algorithm is described as show in Figure 3.

**Figure 3** The pseudo code description of the algorithm RL2W

**Algorithm 3** RL2W algorithm

Input: \(\gamma_p = (g_p, c_p), G = (V, E), u, chi(g, u)\)
Output: \(f(u, v_q)\)
1) \(\text{LCA}(G = (V, E))\)
2) \(\text{if} \ (u \text{ is CR-Mesh Gateway})\{
3) \quad \text{Broadcast } BLM(u, u, \text{level});
4) \} \quad \text{//end if}
5) \(\text{if} \ (u \text{ is CR-Mesh Router})\{
6) \quad \text{while}(\forall g_i \in VG)\{
7) \quad \quad \text{Broadcast } BLM(g_i, u, \text{level});
8) \} \quad \text{//end while}
9) \} \quad \text{//end if}
10) \(\text{if} \ (\text{receive } BLM(g, v, \text{level}))\{
11) \quad \text{if} \ (L(g, u) = \text{level-1}) \{
12) \quad \quad \text{pre}(g, u) = \text{pre}(g, u) \cup \{v\};
13) \} \quad \text{//end if}
5 High reliability and low delay routing algorithm based on DP

With the objective of high reliability and low delay, a distributed routing and spectrum allocation algorithm HRL2A based on dynamic programming is proposed in this section. Firstly, HRL2A computes the channel utility value using CUVC for constructing the high reliability route. Secondly, HRL2A uses the algorithm RL2W computing wireless link weight. Thirdly, the route with high reliability and low latency is constructed based on dynamic programming, and the wireless link channel is allocated.

The process of high reliability and low end-to-end delay route construction based on dynamic programming is the process of solving the optimal value in dynamic programming strategy. The process of channel allocation for wireless link is the process of constructing the optimal solution according to the optimal value in dynamic programming strategy. For the above two processes, a high reliability and low end-to-end delay route construction algorithm DPRA based on dynamic programming and a wireless link spectrum allocation algorithm WLSA are proposed respectively.

5.1 Route construction

$\gamma_p = (g_p, c_p)$ represents a wireless request. $g_p$ denotes the CR-Mesh gateway node, is source node of $\gamma_p$. $c_p$ denotes the CR-Mesh client node, is destination node of $\gamma_p$. We assume that all wireless requests are from the CR-Mesh gateway node to the CR-Mesh client node.

In the algorithm DPRA, the CR-Mesh node is equivalent to each stage of solving optimisation problem based on dynamic programming strategy. The available channel of CR-Mesh node is equivalent to different state of each stage. The state transition equations are given by formula (14) to (16)

\[
dp^*(c_p, c_p, k) = 0 \quad \forall k \in K_{c_p} \quad (14)
\]

\[
dp^*(v_i, c_p, k) = \min_{k' \in K_i} \{ f^k(v_i, v_j) + dp^*(v_j, c_p, k') \} \\
+ dp^*(v_j, c_p, k') \quad \forall k \in K_i \quad (15)
\]

\[
dp(g_p, c_p) = \min_{k \in K_{g_p}} \{ dp^*(g_p, c_p, k) \} \quad (16)
\]

$dp^*(v_i, c_p, k)$ represents the minimum path weight from node $v_i$ to destination node $c_p$ when the node $v_i$ and it is children node is allocated channel $k$. 
Formula (14) for destination node $c_p$, the path weight from destination node $c_p$ to itself on channel $k$ is computed. It is the boundary condition of the dynamic programming strategy.

Formula (15) for CR-Mesh router node $v_i$ to destination node $c_p$, computed on condition that the link between node $v_i$ and its children node $v_j$ is pre allocated channel $k$.

$F(v_i, v_j)$ represents the link weight of $(v_i, v_j)$ allocating channel $k$. $v_j$ represents the children node of node $v_i$ relative to gateway node $g_p$. $dp^*(v_j, c_p, k')$ represents the path weight from CR-Mesh router node $v_j$ to destination node $c_p$, on condition that the link between node $v_j$ and its’ children node is pre allocated channel $k'$.

According to formula (15), on condition that the minimum value of $dp^*(v_i, c_p, k)$ is computed, the children node of node $v_i$, namely, $v_j \in \text{chi}(g_p, v_i)$, and the allocated channel, namely, $k''$, between node $v_j$ and its’ children node are computed for each available channel $k(\forall k \in K_i)$ of node $v_i$. The calculation formula is given by (17)

\[
\begin{align*}
(v_j', k'') &= \arg\min\{dp^*(v_i, c_p, k)\} \quad \forall k \in K_i \\
\forall v_j \in \text{chi}(g_p, v_i), \forall k \in K\{v_i, v_j\} 
\end{align*}
\]  

(17)

Formula (16) for source node of wireless request, the minimum path weight from source node $g_p$ to destination node $c_p$ is computed.

$dp(g_p, c_p, k)$ represents the path weight from CR-Mesh source node $g_p$ to destination node $c_p$, on condition that the link between node $g_p$ and its’ children node is pre allocated channel $k$.

According to formula (16), on condition that the minimum value of $dp(g_p, c_p)$ is computed, the children node of source node $g_p$, namely, $v_j \in \text{chi}(g_p, g_p)$, and the allocated channel, namely, $k''$, between node $g_p$ and node $v_j$, and the allocated channel, namely, $k''$, between node $v_j$ and its’ children node are computed. The calculation formula is given by (18)

\[
\begin{align*}
(v_j', k', k'') &= \arg\min\{dp(g_p, c_p)\} \\
\forall v_j \in \text{chi}(g_p, v_i), \forall k \in K\{v_i, v_j\}
\end{align*}
\]  

(18)

$\arg\min\{dp(g_p, c_p)\}$ represents the calculation of the children node of node $g_p$ and the allocated channel between the node $g_p$ and its’ children node on condition that the minimum value of $dp(g_p, c_p)$ is computed.

$v_j'$ represents children node of node $g_p$, $k'$ represents the allocated channel of link $(g_p, v_j')$, $k''$ represents the allocated channel between $v_j'$ and its children. As shown in Figure 4.

Figure 4 Sketch figure of channel allocation
In the process of calculation path weighted, the route weight packet (RWP) control message is transmitted through common control channel.

RWP($g_p$, $v_i$, $v_q$, $c_p$) represents that the control message sent from node $v_i$ to node $v_q$($v_q \in \text{pre}(g_p, v_i)$), on condition that node $g_p$ is source node, and node $c_p$ destination node for wireless request $\gamma_p$.

RWP($g_p$, $v_i$, $v_q$, $c_p$).RL represents the path weight from CR-Mesh router node $v_i$ to destination node $c_p$.

RWP($g_p$, $v_i$, $v_q$, $c_p$).Ch represents the allocated channel between node $v_i$ and its’ next hop node.

RWP($g_p$, $v_i$, $v_q$, $c_p$).Cn represents the next hop of node $v_i$, on condition that the minimum value of $dp^s(v_i, c_p, k)$ is computed. As shown in Figure 5.

**Figure 5** Sketch figure of the RWP control message

The field calculation formula in RWP is given by (19)

$$
RWP\left( g_p, v_i, v_q, c_p \right) . RL = f^v( v_i, v_j^* ) + dp^s( v_j^*, c_p, k^* )
$$

$$
RWP\left( g_p, v_i, v_q, c_p \right) . Ch = k^*
$$

$$
RWP\left( g_p, v_i, v_q, c_p \right) . Cn = v_j^*
$$

(19)

As for destination node $c_p$, RWP packet fields’ calculation formulae are as follows:

- **RWP($g_p$, $c_p$, $v_q$, $c_p$).RL = 0**, it represents the path weight from CR-Mesh client node $c_p$ to destination node $c_p$ is equal to 0.

- **RWP($g_p$, $c_p$, $v_q$, $c_p$).Ch = 0**, it represents that between node $v_i$ and its’ next hop node is not allocated channel.

- **RWP($g_p$, $c_p$, $v_q$, $c_p$).Cn = $\phi$**, it represents that there are not children node for node $c_p$.

The structure CNV records the information about the path weight from node $v_i$ to node $c_p$, the next hop node of $v_i$, and the pre allocation channel between node $v_i$ and the next hop node of $v_i$. On condition that the node $v_i$ and its’ next hop node is allocated different channel. As shown in Figure 6.
A high reliability and low latency routing algorithm in CWMNs

Figure 6  The structure CNV

Structure 1: CNV structure
1) struct CNV {
2)     NodeValueType dnv; //destination node
3)     NodeValueType nv; //next hop node
4)     ChannelValueType cv; //x(v, nv)
5)     ChannelValueType ncv; //x(nv, nv')
6)     WeightValueType wv; //weight value
7) }

ChannelValueType represents the Channel data type, which ranges from the available channel set. WeightValueType represents the path weight data type. NodeValueType represents the node data type, which ranges from all nodes.

In wireless request γp as for destination node cp, each available channel k(∀k ∈ K) of node vp, a record is saved. Dnv represents destination node cp, nv represents next hop node of node vp, cv represents the allocated channel of link (vi, nv), ncv represents the allocated channel between node nv and its’ next hop node. Wv represents the value of dp*(vi, cp, k).

The pseudo code of DPRA algorithm is described as show in Figure 7.

Figure 7  The pseudo code description of the algorithm DPRA

Algorithm 4 DPRA algorithm
Input: G = (V, E), γp = (gp, cp), u
Output: dp*(vi, cp, k), dp*(gp, cp), CNV
1)   CUVC(X);
2)   RL2W(X);
3) if ((u is destination node ) & (u = = cp)) { 
4)     computing dp*(cp, cp, k) according to (14);
5)   create a new packet RWP(gp, cp, vp, cp);
6)   RWP(gp, cp, vp, cp).RL = dp*(cp, cp, k);
7)   RWP(gp, cp, vp, cp).Ch = 0;
8)   RWP(gp, cp, vp, cp).Cn = Ø;
9) while(∀vp ∈ pre(gp, u)) { 
10)   send RWP(gp, cp, vp, cp) to vp;
11) }  //end while
12) } else if (u is CR-Mesh Router) {
13)   if ( u receive RWP(gp, v, u, cp) ) { 
14)     k’ = RWP(gp, v, u, cp).Ch;
15)     dp*(v, cp, k’) = RWP(gp, v, u, cp).RL;
16)   } 
17) computing dp*(u, cp, k) according to (15);
18) computing (v’, k”) according to (17);
create a CNV record;
CNV.dnv = \( e_p \);
CNV.mv = \( v^* \);
CNV.cv = \( k \);
CNV.ncv = \( k'' \);
CNV.wv = \( dp^* (u, e_p, k) \);
create a new packet \( RWP(g_p, u, v_q, cp) \);
\[ RWP(g_p, u, v_q, cp).RL = dp^*(u, e_p, k) ; \]
\[ RWP(g_p, u, v_q, cp).Ch = k ; \]
\[ RWP(g_p, u, v_q, cp).Cn = v^* ; \]
\[ While(\forall v_q \in pre(g_p, u)) \{ \]
\[ send RWP(g_p, u, v_q, cp) to v_q ; \]
\[ } /end while \]
\[ } /end if \]

5.2 Spectrum allocation algorithm

After the minimum value of \( dp(g_p, e_p) \), which is from CR-Mesh source node \( g_p \) to destination node \( e_p \), is computed by algorithm DPRA. The route is constructed by algorithm WLSA, and the wireless link channel is allocated by algorithm WLSA in this section. This process is the constructing the optimal solution according to the optimal value in dynamic programming strategy.

The channel allocation packet (CAP) control message is transmitted through common control channel from node \( u \) to its’ next hop node \( v \in \chi(h(S_p, u)) \).

\( CAP(g_p, e_p, u, v) \) represents the control message sent from node \( u \) to node \( v \) (\( v \in \chi(h(g_p, u)) \)), on condition that node \( g_p \) is source node, and node \( e_p \) destination node for wireless request \( \gamma_p \).

\( CAP(g_p, e_p, u, v).ch \) represents the allocated channel of link \( (u, v) \). \( CAP(g_p, e_p, u, v).nch \) represents the allocated channel between the node \( v \) and its’ next hop node.

When CR-Mesh router \( v \) received \( CAP(g_p, e_p, u, v) \) control message from the last hop node \( u \), we can summarise the following information:

Firstly, the allocated channel of link \( (u, v) \) is \( CAP(g_p, e_p, u, v).ch \), namely, \( x(u, v) = CAP(g_p, e_p, u, v).ch \).

Secondly, the allocated channel between node \( v \) and its’ next hop node \( v' \) is \( CAP(g_p, e_p, u, v).nch \), namely, \( x(v, v') = CAP(g_p, e_p, u, v).nch \).

Thirdly, a record can be found in CNV records. The record satisfies with that the destination node is \( e_p \), the allocated channel between the node \( v \) and its’ next hop node is
$x(v,v')$. $CAP(g_p, c_p, v, v')$ control message can be constructed and transmitted continually for spectrum allocation.

The pseudo code of WLSA algorithm is described as shown in Figure 8.

**Figure 8** The pseudo code description of the algorithm WLSA

**Algorithm 5** WLSA algorithm

Input: $G = (V, E), \gamma_p = (g_p, c_p), u$

Output: $T_p, x(u, v)$

1) if (u is source node) {
2) computing $(v'_i, k', k'')$ according to (18); 
3) create a new packet $CAP(g_p, c_p, u, v)$;
4) $CAP(g_p, c_p, u, v'_j).ch = k'$;
5) $CAP(g_p, c_p, u, v'_j).nch = k''$;
6) send $CAP(g_p, c_p, u, v'_j)$ to $v'_j$;
7) } else if (u is CR-Mesh Router) {
8) if (u receive $CAP(g_p, c_p, u', u)$) {
9) $x(u', u) = CAP(g_p, c_p, u', u).ch$;
10) $cv = CAP(g_p, c_p, u', u).nch$;
11) for each $nc \in CNV(u)$ {
12) if (nc.drv = $c_p$ & nc.cv = $cv$) {
13) break;
14) } // end if
15) } // end for
16) $v = nc.nv$; //next hop node
17) create a new packet $CAP(g_p, c_p, u, v)$;
18) $CAP(g_p, c_p, u, v).ch = nc.cv$;
19) $CAP(g_p, c_p, u, v).nch = nc.ncv$;
20) send $CAP(g_p, c_p, u, v)$ to $v$;
21) } // end if

6 Simulation and results

In order to validate the efficiency of the algorithms proposed in this paper, we implemented the HRL2A, JCRAS (El-Sherif and Mohamed, 2014), DRC A (Ding and Xiao, 2013) and RCS-DPCS (Mumey et al., 2012) algorithms using NS-2 (Fall and Varadhan, 2001). Minimising the average end-to-end delay is the goal of JCRAS algorithm. Minimising the sum of bandwidths of each session is the goal of DRCA algorithm. Maximising the average throughput is the goal of RCS-DPCS algorithm. High reliability and low end-to-end delay is the objective of HRL2A algorithm.

In the simulation, the network topology region size was $2,000 \times 2,000$ m. The number of nodes in the region are 60, namely, $n = 60$. $T_R = 50$ m, $I_R = 100$ m. The
available number of channels are \(|K| = 8\). The PU uses this channel stochastically. The channel using probability and channel stability randomly selected from the interval \([0, 1]\). The delay in \(ms\) of each channel is a random value in the range \([1, 10]\). The simulated time is 900 s. In a specified simulation time, the number of requests are \(|\Delta| = 40\). The duration in seconds of each wireless request is randomly selected from the interval \([1, 10]\). The average channel using probability is 0.3. The average channel stability is 0.7.

The simulation results that we report are the average of 200 simulation runs. The performance parameters that we report are the average reliability, average end-to-end delay, and average throughput. The average reliability, the average end-to-end delay and the average throughput have confidence interval 90\%, 95\% and 95\%. The average reliability formula is given by (20)

\[ \bar{R} = \frac{1}{|\Delta|} \sum_{\gamma_p \in \Delta} R(\gamma_p) \]  

(20)

\(R(\gamma_p)\) represents the reliability of request \(\gamma_p\), and the formula is given by (21)

\[ R(\gamma_p) = \frac{N(\gamma_p)}{|p(\gamma_p)|} \]  

(21)

\(|p(\gamma_p)|\) represents the number of wireless link in route \(p(\gamma_p)\). \(N(\gamma_p)\) represents the number of failure wireless link in route \(p(\gamma_p)\). The formula is given by (22)

\[ N(\gamma_p) = \sum_{(u,v) \in p(\gamma_p)} fa(u,v) \]  

(22)

\(fa(u,v) = 1\) represents the link \((u,v)\) has been failure in route \(p(\gamma_p)\). \(fa(u,v) = 0\) represents the link \((u,v)\) has not been failure in route \(p(\gamma_p)\).

The average reliability, the average end-to-end delay, the average throughput with different numbers of available channels, channel using probability, channel stability are analysed in the simulation.

### 6.1 Comparison average reliability

We analyse the average reliability of algorithms with different numbers of available channels, channel using probability, channel stability. Figures 9 to 11 show the simulation results.

We can see from Figure 9, the average reliability of HRL2A algorithm is greater than that of the JCRAS, DRCA and RCS-DPCS algorithms. This is because the three algorithms do not consider the route reliability, however, the reliability in the routing metric of HRL2A algorithm. As the number of available channels increase, the average reliability of all four algorithms increases. The average reliability of HRL2A increases more than the other three algorithms.
We can see from Figure 10, the average reliability of HRL2A algorithm is greater than that of the JCRAS, DRCA and RCS-DPCS algorithms. As the channel using probability increases, the average reliability of all four algorithms decreases. This is because the higher the channel using probability used by PU is, the higher link failure probability becomes, which leads to the lower route reliability.

We can see from Figure 11, the average reliability of HRL2A algorithm is greater than that of the JCRAS, DRCA and RCS-DPCS algorithms. As the channel stability increases, the average reliability of all four algorithms increases. This is because the higher the channel stability is, the link failure probability decreases, which leads to the higher route reliability.
6.2 Comparison average end-to-end delay

We analyse the average end-to-end delay of algorithms with different numbers of available channels, channel using probability, channel stability. Figures 9 to 11 show the simulation results.

Figure 12  Average end-to-end delay with different numbers of available channels (see online version for colours)

We can see from Figure 12, the average end-to-end delay of HRL2A and JCRAS algorithms are less than that of the DRCA and RCS-DPCS algorithms. This is because that minimising the average end-to-end delay is the goal of HRL2A and JCRAS algorithms, and minimising the sum of bandwidths of each session is the goal of DRCA algorithm, maximising the average throughput is the goal of RCS-DPCS algorithm. There is no difference between the HRL2A and JCRAS algorithms. As the number of available
channels increase, the average end-to-end delay of all four algorithms decreases. This is because the more the number of available channels can be used, the higher the probability of channel with low latency allocated for route increases, which leads to the lower average end-to-end delay.

We can see from Figure 13, the average end-to-end delay of HRL2A and JCRAS algorithms are less than that of the DRCA and RCS-DPCS algorithms. As the channel using probability increases, the average end-to-end delay of all four algorithms increases. This is because the higher the channel using probability used by PU is, the higher link failure probability becomes, which leads to the higher average end-to-end delay.

**Figure 13** Average end-to-end delay with different channel using probability (see online version for colours)

![Figure 13](image)

**Figure 14** Average end-to-end delay with different channel stability (see online version for colours)

![Figure 14](image)
We can see from Figure 14, the average end-to-end delay of HRL2A and JCRAS algorithms are less than that of the DRCA and RCS-DPCS algorithms. As the channel stability increases, the average end-to-end delay of all four algorithms decreases. This is because the higher the channel stability becomes, the link failure probability increases, which leads to the lower average end-to-end delay.

6.3 Comparison average throughput

We analyse the average throughput of algorithms with different numbers of available channels, channel using probability, channel stability. Figures 15 to 17 show the simulation results.

We can see from Figure 15, the average throughput of HRL2A and RCS-DPCS algorithms are greater than that of the JCRAS and DRCA algorithms. This is because that the HRL2A algorithm with high reliability and low delay as the goal, the higher the reliability is, the lower the wireless link failure probability is, which leads to the higher average throughput. Maximising the average throughput is the goal of RCS-DPCS algorithm. Minimising the sum of bandwidths of each session is the goal of DRCA algorithm. Minimising the average end-to-end delay is the goal of JCRAS algorithms. The average throughput of the RCS-DPCS algorithm is slightly higher than that of HRL2A algorithm, and the average throughput of the DRCA algorithm is slightly higher than that of JCRAS algorithm. As the number of available channels increase, the average throughput of all four algorithms increases. This is because the number of available channels, available network resources increase, network throughput, which leads to the higher average throughput.

*Figure 15* Average throughput with different numbers of available channels (see online version for colours)
A high reliability and low latency routing algorithm in CWMNs

We can see from Figure 16, the average throughput of HRL2A and RCS-DPCS algorithms are greater than that of the JCRAS and DRCA algorithms. As the channel using probability increases, the average throughput of all four algorithms decreases. This is because the higher the channel using probability used by PU is, the higher link failure probability becomes, which leads to the lower average throughput delay.

Figure 17  Average throughput with different channel stability (see online version for colours)

We can see from Figure 17, the average throughput of HRL2A and RCS-DPCS algorithms are greater than that of the JCRAS and DRCA algorithms. As the channel stability increases, the average throughput of all four algorithms increases. This is because the higher the channel stability is, the link failure probability decreases, which leads to the higher average throughput.
7 Conclusions

The problem of routing and spectrum allocation with the goal of high reliability and low end-to-end delay is researched in this paper. A CUVC algorithm is proposed. A high reliability and low latency wireless link weight computing algorithm (RL2W) is proposed. On this basis, a high reliability and low latency routing and spectrum allocation algorithm based on dynamic programming in CWMNs (HRL2A) is proposed.

Simulation results show that HRL2A algorithm can achieve higher reliability and low average end-to-end delay. The average reliability of HRL2A algorithm is higher than that of JCRAS, DRCA and RCS-DPCS algorithm. The average end-to-end delay is lower than that of DRCA and RCS-DPCS algorithm. There is no difference between the HRL2A and JCRAS algorithms. The JCARS algorithm is also considered for the optimal algorithm with the optimisation of minimising the average end-to-end delay. Therefore, we can derive that HRL2A algorithm can achieve higher reliability, and can achieve low average end-to-end delay at same time. Which leads to that HRL2A algorithm can achieve higher the average throughput. Routing and spectrum allocation problem with more goals is researched in the future.

Acknowledgements

The authors would like to thank the reviewers for their detailed comments that have helped to improve the quality of the paper, and thank Software engineering Postdoctoral Station of Central South University for supporting research environment. This work is supported by National Natural Science Foundation of China under Grants No. 61309027, 61272151, and Scientific Research Fund of Hunan Provincial Education Department No. 13B148, and supported by China Postdoctoral Science Foundation No.2013M542136, and supported by Hunan Province postdoctoral research Foundation No. 2014RSA4027, and supported by the Postdoctoral foundation of Central South University No.126224, and supported by Youth Scientific Research Foundation of Central South University of Forestry and Technology No. QJ2011002A.

References


