Joint Routing and Channel Assignment for Delay Minimization in Multi-Channel Multi-Flow Mobile Cognitive Ad Hoc Networks

Tong Zhou†, Feilong Tang†, Jie Li‡, Yanqin Yang♮, Wenchao Xu♮, Bin Yao⋆, Minyi Guo⋆
† School of Software, Shanghai Jiao Tong University, Shanghai, China
‡ Faculty of Engineering, Information and Systems, University of Tsukuba, Japan
♮ School of Information Science Technology, East China Normal University, Shanghai, China
⋆ Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
zhoutong@sjtu.edu.cn, {tang-fl,guo-my,yaobin}@cs.sjtu.edu.cn, lijie@cs.tsukuba.ac.jp, {yqyang, wchxu}@ce.ecnu.edu.cn

***Abstract*—Channel interference and node mobility cause significant performance degradation to wireless networks. In multi-channel multi-flow mobile cognitive ad hoc networks, it becomes even worse due to both unexpected primary user activities and potential interference among multiple flows. In this paper, we propose a *Joint Routing and Channel Assignment (JRCA)* approach based on delay prediction. Firstly, it formulates the JRCA problem with the objective of delay minimization. Next, a delay prediction model is proposed based on the channel collision probability. Then, a heuristic algorithm joints routing and channel assignment is designed to solve the JRCA problem. the JRCA algorithm can find out the path with minimal end-to-end (e2e) delay. NS2-based simulation results demonstrate that the JRCA approach significantly outperforms related proposals in terms of average e2e delay.**

Keywords*—*Cognitive radio network, end-to-end delay, routing and channel allocation

I. Introduction

Cognitive radio network has been proposed as a promising networking paradigm to alleviate the severe scarcity in unlicensed spectrum and to improve the efficiency of licensed spectrum utility. In mobile cognitive ad hoc networks (*M-CADN*), secondary users (SU) opportunistically access the idle licensed channels of primary users (PU) through frequently sensing PU activities and immediately vacating the spectrum when the corresponding PU activates again. In general, each time slot starts with a sensing period, and then a transmission period.

The unexpected activation of PUs and the node mobility cause serious challenges to delay-sensitive applications in MCADNs. In multi-channel multi-flow MCADNs, it becomes even worse because multiple links potentially interfere with each other. To minimize the delay requirement, essentially, it is important to address the following key issues. Firstly, routing and channel assignment should be jointly designed, because available channels of SUs are location- and time-dependent on PUs activation. As shown in Fig.1, there are eight secondary users $SU_s, SU_1 - SU_6, SU_d$ and three primary users PU_1, PU_2 and PU_3 in a MCADN. At this moment, the PU_1, PU_2 and PU_3 are sending packets using their licensed channels c_1, c_2 and c_3, respectively. Simultaneously, a delay-sensitive flow f_k needs to be transmitted from the source node SU_s to the destination node SU_d. There are three path candidates for f_k: $P_1^{f_k}$=SU_s→SU_1→SU_2→SU_d and $P_2^{f_k}$=SU_s→SU_4→ SU_6→SU_d and $P_3^{f_k}$=SU_s→SU_3→SU_5→SU_d. Since secondary users should not interfere any primary users, SU_2 has no available channel currently. As a result, $P_1^{f_k}$ can not be selected for f_k.

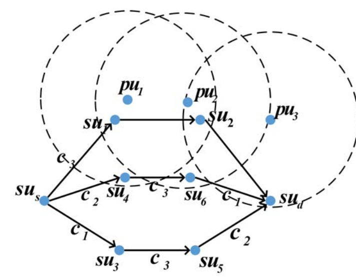

Fig. 1 Motivation scenario for our JRCA.

Secondly, it is indispensable to set up a delay estimation model to capture the co-channel interference among PUs and SUs. Once again, for the above two candidate paths $P_2^{f_k}$ and $P_3^{f_k}$, we need an accurate delay model to decide which path is better in terms of end-to-end (e2e) delay.

Cross-layer design has been studied for wireless networking in recent years for throughput optimization [1] [2] and so on. Some schemes proposed effective transmission time (ETT) [3] [4], spectrum switching delay [5] to control e2e delay, based on cross-layer interactions.

Based on the above analysis, it has been shown that channel collision probability is a key point for e2e delay. In this paper, the joint routing and channel assignment (JRCA) algorithm was proposed for MCADNs. Besides, the main investigation of the paper is concentrated on channel collision predication which is widely used by the e2e delay metric. Main contributions of this paper are summarized as follows.

- Formulating the JRCA problem to minimize e2e delay in MCADNs.
- Proposing an e2e delay model, which consists of transmission time and media access time. Using the channel conflict probability, we model *expected transmission time* (ETT) and *expected media access time* EMAT to quantitively predict e2e delay.
- Proposing a novel routing metric based on the ETT and EMAT to capture channel interference among PUs and

SUs

- Based on the proposed routing metric, this paper designed a distributed JRCA algorithm for delay minimization based routing and channel allocation in MCADNs.

The rest of this paper is organized as follows. Section II reviews main related works. Section III describes the system model of MCADNs and formulates the JRCA problem. Section IV presents a delay prediction model based on the collision probability, and proposes an e2e delay based routing metric. Section V presents a new distributed JRCA algorithm built on proposed routing metric. In Section VI, we evaluated our JRCA algorithm by comparing it with related proposals STODRP and WCETT, using NS2 based simulation system. Finally, section VII concludes this paper along with a discussion on future work.

II. Related Work

This section reviews related work about cross-layer design and e2e delay in wireless cognitive networks.

Cross-layer design has been widely studied in recent years, which requires a tight coupling between the routing and the spectrum management [1] [2]. In [1], authors pointed out that network throughput is a function of time and spectrum, and estimated the spectrum using time, which secondary users could get from primary users, so that secondary users could assign channels better. A new conception: spectrum utility was proposed in [2] as well as a new algorithm which is aimed to maximize throughput by jointly considering routing, spectrum allocation and power control.

In addition to the above related works, many classical e2e delay routing metrics capture factors take both routing and channel assignment into consideration. In [6] authors used the hop number to be the metric of the routing algorithm and modified the ad-hoc distance vector (AODV) routing protocol for cognitive networks. However, these schemes can not estimate e2e path delay properly. A new metric that takes spectrum switching delay and backoff delay into account was proposed in [5]. Moreover, Effective Transmission Time (ETT) metric [3] captured the transmission delays on links as $\frac{L}{T(1-p)}$. It modeled the transmission time by taking the expected number of retransmissions into account. In STODRP [4], it used a routing metric combined transmission delay, channel switching delay and protocol delay. In all these delay oriented related works, channel collision probability played an important role to estimate delay exactly.

III. System Model and Problem Formulation

Considering a MCADN consists of M primary users that hold licensed spectrum bands and can occupy their assigned channel arbitrarily, and N secondary users that opportunistically send their data using idle channels of primary users.

A MCADN is modeled as an directed graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ is the union of the SU set (V_S) and the PU set (V_P) such that $\mathcal{V}$=V$_S \bigcup$V$_P$, with $\|V_S\|$=N and $\|V_P\|$=M. $\mathcal{E}$ is the union of E_S (the set of links among SUs) and E_P (the set of links among PUs) such that E=$E_S \bigcup E_P$. R_T and R_I denote the transmission and interference range respectively.Any $e_{v,u} \in \mathcal{E}$ represents a directional link from v to u, where $\| v-u \| \leq R_T$.

Each $v \in V_S$ is equipped with q cognitive radios, which are able to detect available data channels [7] [8]. Each SU is also equipped with a traditional wireless interface, which forms a common control channel (CCC) to transmit control messages (such as RREQ in section V). The CCC does not interfere with any data channels. In this paper, assuming that all nodes use an identical transmit power, so that node v is in node u's transmission range implies that node u is also in node v's transmission range. Let $\mathcal{DC}$ be a set of L orthogonal data channels in a MCADN, i.e., $\mathcal{DC} = \{c_1, \ldots, c_L\}$, and each channel $c_i \in \mathcal{DC}$ be assigned an identical bandwidth.

In this paper, the effect of the interference among different nodes is modeled as that simultaneous packet transmissions from interfering nodes result in the loss of all involved packets. Two nodes v and u interfere with each other if they use the same channel and $\| u-v \| \leq R_I$. Available channel set $\mathcal{AC}_v$ of a SU v changes with time due to the arbitrary appearance of PUs and node mobility. So, time is divided as a series of discrete time slots and assuming that $\mathcal{AC}_v$ and network topology keep fixed within any single time slot. Consequently, the available channel set of a link $e_{v,u} \in \mathcal{E_S}$ in a slot t is $\mathcal{AC}_{v,u}(t)$=$\mathcal{AC}_v(t) \bigcap \mathcal{AC}_u(t)$.

TABLE I Notations

Notations	Meaning
$e_{v,u}$	a link from SU v to u
$\mathcal{AC}_v(t)$	the set of available channels of a SU v at a slot t
f_k	the kth data flow f_k=(s_k, d_k) ($1 \leq k \leq K$), where s_k and d_k are source and destination nodes of f_k
$\mathcal{I}_v$	the primary node set of PUs interfered by a SU v
$D^{f_k}_{e^c_{v,u}}$	delay of f_k over a link $e_{v,u}$ and a channel c
$\mathcal{N}_v$	neighbor nodes set of node v

IV. Delay Prediction

The basic principle of the routing protocol is to minimize e2e delay through assigning interfering links with orthogonal channels. This section presents how to predict the delay of a link-channel $e^c_{v,u}$, and then formulates e2e delay of a path, which will be used as a routing metric in the next section.

In MCADNs, a link delay $D^{f_k}_{e^c_{v,u}}$ are mainly caused by *media access time (MAT)* and *transmission time (TT)*. MAT refers to the waiting time before a SU actually transmits a data packet and TT is the time from transmitting a data packet to receiving an ACK message. As a result, $D^{f_k}_{e_{v,u}}$ can be calculated using the following formula.

$$D^{f_k}_{e^c_{v,u}} = EMAT^c_{v,u} + ETT^c_{v,u} \quad (1)$$

where $EMAT^c_{v,u}$ and $ETT^c_{v,u}$ are expected media access time and expected transmission time of a link-channel $e^c_{v,u}$, respectively.

A. Expected Transmission Time

In MCADNs, co-channel interference among neighboring links will cause packet retransmission, which increases the e2e delay and decreases the network throughput. Since it is impossible to get accurate media access time and transmission time in dynamical MCADNs, the expected transmission time is used to measure the packet transmission delay.

Assuming that $P^c_{v,u}$ be the collision probability of a link-channel $e^c_{v,u}$ and $ETX^c_{v,u}$ be the expected number of times that a given packet is transmitted from v to u over $e^c_{v,u}$. $ETT^c_{v,u}$ and $ETX^c_{v,u}$ can be calculated using formulas (2) and (3) respectively.

$$ETT^c_{v,u} = ETX^c_{v,u} \times (T^{c,DATA}_{v,u} + T^{c,ACK}_{u,v}) \quad (2)$$

$$ETX^c_{v,u} = \frac{1}{1-P^c_{v,u}} \tag{3}$$

where $T^{c,ACK}_{u,v}$ is the time of transmitting an ACK packet over a link-channel $e^c_{v,u}$, and $T^{c,DATA}_{m,n}$ is the time of transmitting a data packet over a link-channel $e^c_{v,u}$, which can be obtained as formular (4).

$$T^{c,DATA}_{v,u} = \frac{S_{packet}}{B^c} \tag{4}$$

S_{packet} is the size of a data packet and B^c is the channel capacity of channel c on the link $e_{v,u}$. $T^{c,ACK}_{v,u}$ can be calculated similarly.Following is the predication of the collision probability of a link-channel $e^c_{v,u}$.

As shown in formula (3), $ETX^c_{v,u}$ depends on the collision probability $P^c_{v,u}$ of the link-channel $e^c_{v,u}$. In MCADNs, channel collision falls into the following three categories, as shown in Fig. 2, where node v is in the node u's transmission range, nodes M_1, M_2, M_3 and N_3 are in node u's interference range.

1) *Data packet collision.* A SU M_1 sends data packets to N_1, and meanwhile v sends data packets to u. Data packets from M_1 and data packets from v cause collision at u. The collision probability is denoted as $P1^c_{v,u}$. 2) *ACK packet collision.* A SU N_2 sends data packets to M_2 and M_2 responds ACK packets to N_2. Meanwhile, v sends data packets to u. In this case, ACK packets from M_2 and data packets from v cause collision at u. The collision probability is denoted as $P2^c_{v,u}$. 3) *Data and ACK collision.* A SU M_3 sends data packets to N_3 and N_3 sends ACK packets to M_3. Meanwhile, v sends data packets to u. Consequently, the ACK packets from N_3 and data packets from M_3 cause collision with data packets from v. The collision probability is denoted as $P3^c_{A,B}$.

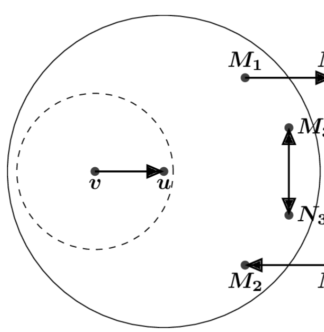

Fig. 2 Three categories of collisions.

Assuming that each node sends packets follow the Poisson distribution. A link-channel $e^c_{m,n}$ sends data packets at a rate $\lambda^c_{m,n}$, and sends ACK packets at a rate $\gamma^c_{m,n}$. The probability $P^c_{m,n}(X=0)$ of the link-channel $e^c_{m,n}$ can be solved in the formulas (5) (6). $P^c_{m,n}(X=0)$ refers to the probability of the link-channel $e^c_{m,n}$ sending 0 packet in a unit time. As a result, the probability $P^c_{m,n}(X=0, t=T)$ of $e^c_{m,n}$ sends 0 packet during T slots can be formulated in formulas (7) (8).

$$P^{c,DATA}_{m,n}(X=0) = exp\{-\lambda^c_{m,n}\} \tag{5}$$

$$P^{c,ACK}_{m,n}(X=0) = exp\{-\gamma^c_{m,n}\} \tag{6}$$

$$P^{c,DATA}_{m,n}(X=0, t=T) = (exp\{-\lambda^c_{m,n}\})^T \tag{7}$$

$$P^{c,ACK}_{m,n}(X=0, t=T) = (exp\{-\gamma^c_{m,n}\})^T \tag{8}$$

Without loss of generality, supposing that v starts sending data packets to u at a time slot t_1, as shown in Fig.3. If M_1 sends data packets to N_1 during time interval $(0, t_2]$, data packets from v to u and data packets from M_1 to N_1 will conflict at u. Otherwise, no collision occurs. $P1^c_{v,u}$ can be calculated with formula (9).

$$\begin{aligned} P1^c_{v,u} &= 1 - P^{c,DATA}_{M_1,N_1}(X=0, t = T^{c,DATA}_{M_1,N_1} + T^{c,DATA}_{v,u}) \\ &= 1 - exp\{-\lambda^c_{M_1,N_1}(T^{c,DATA}_{M_1,N_1} + T^{c,DATA}_{v,u})\} \end{aligned} \tag{9}$$

Similarly, $P2^l_{v,u}$ and $P3^3_{v,u}$ can be calculated with formulas (10) (11).

$$P2^c_{A,B} = 1 - exp\{-\gamma^c_{M_2,N_2}(T^{c,ACK}_{M_2,N_2} + T^{c,DATA}_{v,u})\} \tag{10}$$

$$\begin{aligned} P3^c_{v,u} = 1 - exp\{-\{&\lambda^c_{M_3,N_3}(T^{c,DATA}_{M_3,N_3} + T^{c,DATA}_{v,u}) \\ &+ \gamma^c_{M_3,N_3}(T^{c,ACK}_{M_3,N_3} + T^{c,DATA}_{v,u})\}\} \end{aligned} \tag{11}$$

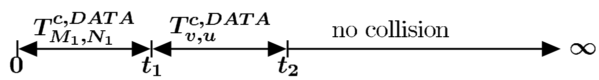

Fig. 3 Data packet collision.

Assuming that channel collisions are independent with each other so that $P^c_{v,u}$ can be calculated using formula (12).

$$\begin{aligned} P^c_{v,u} = 1 - &\prod_{e^c_{m,n} \in \mathcal{I}^1(e^c_{v,u})} exp\{-\lambda^c_{m,n}(T^{c,DATA}_{m,n} + T^{c,DATA}_{v,u})\} \\ &\times \prod_{e^c_{m,n} \in \mathcal{I}^2(e^c_{v,u})} exp\{-\gamma^c_{m,n}(T^{c,ACK}_{m,n} + T^{c,DATA}_{v,u})\} \\ &\times \prod_{e^c_{m,n} \in \mathcal{I}^3(e^c_{v,u})} exp\{-\{\lambda^c_{m,n}(T^{c,DATA}_{m,n} + T^{c,DATA}_{v,u}) \\ &+ \gamma^c_{m,n}(T^{c,ACK}_{m,n} + T^{c,DATA}_{v,u})\}\} \end{aligned} \tag{12}$$

where $\mathcal{I}^k(e^c_{v,u})(k = 1,2,3)$ denotes the set of three kinds of interfering links.

B. Expected Media Access Time

In MCADNs, all nodes are synchronized; and every transmission starts at the beginning of a slot. When a node attempts to send a packet, it needs to wait for some time slots which is randomly selected from $\{0, 1, \dots, W_0 - 1\}$, where W_0 is an integer representing the initial contention window size. Whenever a collision occurs, the contention window size increases r times through the backoff algorithm so that the collision probability in the next transmission will significantly be decreased. So the total number of waiting time slots can be used to estimate the expected media access time. Based on [9], using the following formula to calculate the EMAT.

$$\overline{D_{\sum}}^c_{v,u} = \frac{1}{2}\left(\frac{1}{1-p^c_{v,u}} + \frac{W_0}{1 - r \times p^c_{v,u}}\right) - 1 \tag{13}$$

$$EMAT^c_{v,u} = \overline{D_{\sum}}^c_{v,u} \times \text{slot} \tag{14}$$

where $\overline{D_{\sum}}^c_{v,u}$ is the average number of waiting time slots for a link-channel $e^c_{v,u}$; $p^c_{v,u}$ is the collision probability of a link-channel $e^c_{v,u}$; *slot* is the length of a time slot; and r is the expand factor of w_0.

C. Expected e2e Path Delay

According to IV-A and IV-B, the link delay in formula (1) can be calculated in terms of collision probability. Now, it is easy to calculate the e2e path delay $D^{f_k}_{path}$ for a flow f_k by accumulating the delay of links involved in the path, formulated in (15).

$$D^{f_k}_{path} = \sum_{e^c_{v,u} \in f_k} D^{f_k}_{e^c_{v,u}} \tag{15}$$

V. Joint Routing and Channel Assignment Based on Minimal Delay

This section shows the Joint Routing and Channel Assignment (JRCA) protocol, using the e2e path delay $D_{path}^{f_k}$ in (15) as the routing metric. In JRCA, each cognitive node makes decisions on route selection and spectrum allocation based on local information.

A. Link Stability Prediction

Besides the channel interference, In MCADNs, the duration of a link $e_{v,u}$ also significantly suffers from relative movement of v and u and PU activation. So we need to predict maximal lifetime $MLT_{v,u}^c$ of a link-channel $e_{v,u}^c$ due to the node mobility. If $MLT_{v,u}^c < D_{e_{v,u}^c}^{f_k}$, $e_{v,u}$ should not be involved in the route.

Assuming that node mobility follows Random Waypoint (RWP) model. Based on previous work, when the distance $d_{v,u}(t)$ between v and u equals R_T, t reaches the *mobility duration* $T_{v,u}$, which can be easily solved by through setting $d_{v,u}(t) = R_T$. Similarly, by setting $d_{v,m}(t) = R_I$, the longest duration T_{v,PU_m}^c can be calculated. T_{v,PU_m}^c measures how long node v using channel c will interfere with the PU_m, where PU_m is using its channel c. Note that it is supported by the fact that different PUs use different licensed channels in a MCADN. Then $MLT_{v,u}^c$ can be solved through the formula (16).

$$MLT_{v,u}^c = min\{T_{v,u}, T_{v,PU_m}^c, T_{u,PU_m}^c\} \tag{16}$$

Then, $MLT_{v,u}^c$ is the maximal time period that a currently available link can keep if no change in velocities happens. The $P(MLT_{v,u}^c)$ (shown in formula (17)) captures possible changes in velocities that may happen during the period $MLT_{v,u}^c$. Finally, the expected maximal lifetime $EMLT_{v,u}^c$ of $e_{v,u}^c$ can be predicted with the formula (18), which considers random changes of nodes' speeds and directions [10].

$$P(MLT_{v,u}^c) \approx e^{-\lambda MLT_{v,u}^c \times e^{-\lambda\tau}} + \zeta(1 - e^{-\lambda MLT_{v,u}^c}) \tag{17}$$

where λ^{-1} is the mean epoch of nodes; and τ and ζ can be estimated by measurement.

$$EMLT_{v,u}^c = MLT_{u,u}^c \times P(MLT_{u,u}^c) \tag{18}$$

B. JRCA Protocol

In the JRCA protocol, each node keeps a table to record the collision probability of its available channels. In the initial period, the collision probability of each available channel is set to 0. After sending packets for a period, each node can record the collision probability of parts of available channels by which the node sent packets. For the other available channels, which have not been used to send packets, cognitive nodes within the interference range can use hello packets to keep communicating with neighbor nodes such as exchanging channel information (e.g., rate of sending data packets and ACK packets on the channel) and calculating collision probability using formulas (12) (19). So v's collision probability p_v^c on each available channel c can be calculated using formula (19).

$$p_v^c = 1 - \prod_{u \in \mathcal{N}_v} (1 - p_{v,u}^c) \tag{19}$$

where $\mathcal{N}_v$ denotes set of neighbor nodes of v.

1) Joint path selection and channel assignment: Initially, the source node s_k broadcasts a RREQ (routing request) packet with the format shown in table II, where each node is assumed have 12 channels. On receiving a RREQ packet, a relay node performs JRCA algorithm, which will be run hop by hop until reaching destination d_k in a distributed fashion, as shown in Algorithm 1.

TABLE II RREQ message format

type	reserved	hop count	RREQ id
Destination address		Originator address	
Destination sequence number		Originator sequence number	
total delay			
channelClsPro[]			
node address	link channel		link delay

Algorithm 1: Joint routing and channel assignment

Input: RREQ packet p for a flow f_k
if $(v = rq.src)$ **then**
  drop p
  return
end if
minDelay = ∞
minChannel = 0
for $(\forall c \in \mathcal{AC}_v \wedge \mathcal{AC}_u)$ **do**
  calculate $D_{e_{u,v}^c}^{f_k}$ using (1)
  calculate $EMLT_{u,v}^c$ using (18)
  if $(EMLT_{u,v}^c > p.delay + D_{e_{u,v}^c}^{f_k}) \wedge$ $(D_{e_{u,v}^c}^{f_k} \leq minDelay)$ **then**
    minDelay = $D_{e_{u,v}^c}^{f_k}$
    minChannel = c
  end if
end for
p.delay += minDelay
get reverse route rt_{src} from v's routing table
get route rt_{dest} from v's routing table
if $(rt_{src} = NULL) \vee (rt_{src}.delay \geq p.delay)$ **then**
  update rt_{src}
else
  return
end if
if $(v = p.rp\_dest) \vee (rt_{dest} \neq NULL)$ **then**
  replay a RREP packet with rt_{src}
else
  broadcast p
end if

During route setup period, cognitive nodes use JRCA protocol to carry out its local optimal choice hop by hop towards a destination node. As shown in Algorithm 1, when a node v receives a RREQ packet for a flow $f_k = (s_k, d_k)$ from its upstream node u, it deals with the RREQ packet as follows.

1) v checks if it is s_k. If $v = s_k$, it drops the RREQ packet. 2) v uses the RREQ packet information calculating link delay $D_{e_{u,v}^c}^{f_k}$ for each available channel c of link $e_{u,v}$. If $EMLT_{u,v}^c > p.delay + D_{e_{u,v}^c}^{f_k}$, which means that the link-channel $e_{u,v}^c$ can be used in the link maximal lifetime. And $p.delay$ means the total delay from source node to the node u. Then node v assigns a channel with minimal link delay (minDelay) for the link $e_{u,v}$. 3) v accumulates the minDelay

to the delay recorded in RREQ packet $p.delay$. Now $p.delay$ refers to transmission time from s_k to v. If the expected delay is smaller than the delay in the route table, node v will replace the old reverse route with the new reverse route from source s_k to node v. 4) If v has a route to destination in its route table or v is the destination d_k , node v makes a RREP packet and send this RREP packet along with current reverse route. Otherwise, it will broadcast RREQ. The format of the RREP (routing reply) packet is shown as table III.

Algorithm 1 chooses the best link-channel $e^c_{u,v}$ hop by hop. Finally, JRCA gets the minimal e2e delay path as well as channel assignment from source to destination.

TABLE III RREP message format

type	reserved	hop count
RREQ id		
Destination address	Destination sequence number	
Originator address	Originator sequence number	
total delay		

Algorithm 2: Route setup

Input: RREP packet p for flow f_k

get route rt_{dest} from route table
get reverse route rt_{src} from route table
if $((u = p.rp_{src}) \wedge (p.delay < rp_{src}.delay))$ **then**
  update rt_{dest}
else
  if $(rt_{dest} = NULL)$ **then**
    insert a new rt_{dest} into route table
  end if
  if $(p.delay - rt_{src}.delay < rt_{dest}.delay)$ **then**
    update rt_{dest}
  end if
end if
forward p with the route rt_{src}

2) Route set up: When a node u receives a RREP packet from a node v, it deals with the RREP packet using algorithm 2 with the following steps.

1) u gets the route to the source node rt_{src} (i.e., s_k) and the route to the destination node rt_{dest} (i.e., d_k) from its route table. Note that rt_{dest} might be null. 2) If u is the source node and the delay in RREP packet is lower than that in rt_{dest}, u updates its route table. 3) If u is not the source node, it updates its route to rt_{dest} in route table based on the RREP packet. And u forwards this RREP packet to the next hop using rt_{src}.

C. local route repair

When a route fails at the link-channel $e^c_{v,u}$, v broadcasts a Route Error packet (includes the disabled link-channel $e^c_{v,u}$, destination d_k and collision probability of v's available channels) to neighbors. When a neighbor $n_v \in \mathcal{N}_v$ receives the Route Error packet, n_v performs local route repair as follows: 1) n_v checks its route table. If there is a route is using the link $e_{v,u}$ and the channel c, n_v removes this route in its route table. 2) If there is still an active route to the destination node in n_v's route table, n_v replies a Error Response packet (include minimal link-channel delay $D^{f_k}_{e^c_{v,n_v}}$, path $P^{f_k}_{n_v,d_k}$ from n_v to destination node d_k and the e2e delay of $P^{f_k}_{n_v,d_k}$) to node v. 3) Whenever node v receives an Error Response packet, if there is no usable route to destination or new e2e delay is better than old one, node v updates its route table and sends this route to the source node with reverse route. 4) If node v doesn't receive any Error Response packet from its neighbors, v notifies the upstream node w. Node w does the same local route repair as node v. 5) If Route Error packet reaches source node s_k, s_k will make a route request packet once more and use JRCA protocol get the minimal e2e delay path.

VI. Simulation And Evaluation

This section developed a simulation system, which was built on the NS2 simulator with multi-channel extensions. Firstly, table IV describes the simulation system setting. Then, the simulation system comprehensively evaluated the JRCA by comparing it with some related proposals in terms of various performance metrics.

TABLE IV System Parameters

Parameters	**Value**
Network Size	1500*1500 m^2
Number of SUs	60
Simulation Time	500 s
Number of data flows	8
Number of channels	8
Transmission range	125 m
Interference range	250 m
Initial content window (w_0)	256
Length of a slot	50 us
Maximal speed	10 m/s

In the system, PUs and SUs are randomly deployed in an area of 1500m×1500m. Both SUs and PUs move at a speed randomly distributed in [0,V_{max}]. Signal propagation was set as Two-Ray Ground Reflection model. CNs and PNs were set the same transmission radius and interference range. Each PU was assigned a fixed data channel and can randomly use it.

A. Performance Evaluation

The system evaluated the JRCA protocol by comparing it with the well-known cognitive network routing protocols.

- *STODRP (Spectrum-Tree base On-Demand routing protocol)* [4] the ETT (expected transmission time), protocol delay and channel access delay as routing metric. Besides, STODRP builds a spectrum tree ST^c for each available channel c to help SUs routing and allocate channels. STODRP adds all the nodes which have available channel c into ST^c.
- *WCETT (Weighted cumulative Expected Transmission Time)* [11]. WCETT is also a delay oriented routing metric. WCETT just establishes routing path as AODV, and it takes ETT and channel interference along a flow into consideration.

1) E2e delay with channel number: The number of channels significantly affects e2e performance. As shown in Fig.4, as more and more channels were added into the network, e2e delay and average throughput respectively decreases and increases step by step in the three proposals. But the JRCA always exhibits the best delay among them. The more channels, however, the more spectrum trees so that the spectrum tree algorithm will become more and more complex at each root node.

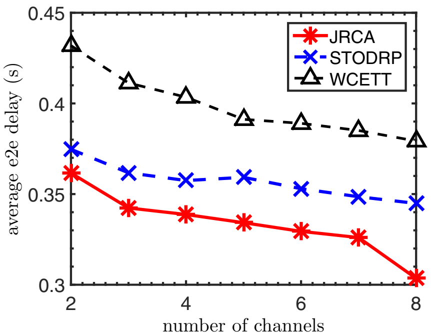

Fig. 4 E2e delay with the number of channels.

2) E2e delay with flow number: In a given network, more flows will cause more serious channel conflict. From Fig.5, it is easy to find that as data flows increase, average e2e delay increases in all proposal. However, the JRCA always exhibits the better performance than the STODRP and WCETT. Both JRCA and STODRP consider channel collision among different routes but WCETT don't take it into account so that the WCETT has a higher delay than other two proposals.

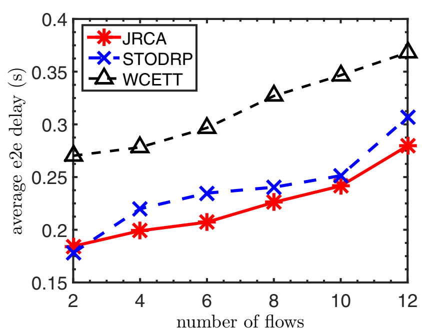

Fig. 5 E2e delay with the number of data flows.

3) E2e delay with speed: Using different maximal speeds to test three routing algorithms. As shown in Fig.6, with the maximal speed increases, e2e delay increases. The reason is that the JRCA considers node mobility but WCETT and STODRP did not take this consideration into account. So path duration in WCETT and STODRP is lower than that in the JRCA. As a result, the JRCA exhibits the best e2e delay, shown in Fig.6.

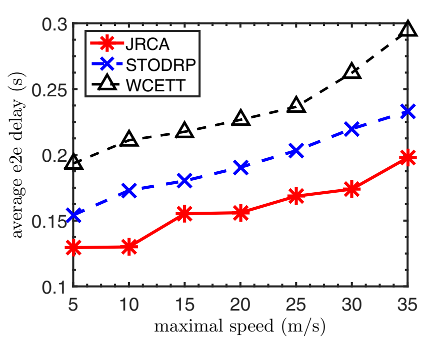

Fig. 6 E2e delay with the maximal speed.

VII. Conclusions and Future Work

In this paper, a JRCA protocol that jointly selects routes and assigns channels was designed based on the delay prediction in MCADNs. Firstly, it proposed an e2e delay model, which consists of the expected media access time and expected transmission time, to predict e2e delay according to the channel collision probability. Then a heuristic routing algorithm was implemented so that the JRCA protocol jointly explored routes with the minimal e2e delay and assigns channels for MCADNs. Finally, a NS2-based simulation system demonstrated that the JRCA protocol significantly outperform related proposals in terms of average e2e delay.

VIII. Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) project (Nos. 61373156, 91438121, 61202025 and 61428204), the Key Basic Research Project (No. 12JC1405400) and the Shanghai Pujiang Program (No. 13PJ1404600) of the Shanghai Municipality, the National Basic Research Program (973 Program, No. 2015CB352403), and Shanghai Branch of Southwest Electron and Telecom Technology Research Institute Project (No. 2013008).

References

[1] I. Pefkianakis, S. Wong, and S. Lu, "Samer: Spectrum aware mesh routing in cognitive radio networks," in *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd*, Oct 2008, pp. 1–5.

[2] L. Ding, T. Melodia, S. Batalama, J. Matyjas, and M. Medley, "Cross-layer routing and dynamic spectrum allocation in cognitive radio ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1969–1979, May 2010.

[3] H.-P. Shiang and M. van der Schaar, "Distributed resource management in multihop cognitive radio networks for delay-sensitive transmission," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 2, pp. 941–953, Feb 2009.

[4] G.-M. Zhu, I. Akyildiz, and G.-S. Kuo, "Stod-rp: A spectrum-tree based on-demand routing protocol for multi-hop cognitive radio networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, Nov 2008, pp. 1–5.

[5] G. Cheng, W. Liu, Y. Li, and W. Cheng, "Spectrum aware on-demand routing in cognitive radio networks," in *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd*, April 2007, pp. 571–574.

[6] K. Chowdhury and M. Di Felice, "Search: A routing protocol for mobile cognitive radio ad-hoc networks," in *Sarnoff Symposium, 2009. SARNOFF '09. IEEE*, March 2009, pp. 1–6.

[7] H. Li, X. Cheng, K. Li, C. Hu, N. Zhang, and W. Xue, "Robust collaborative spectrum sensing schemes for cognitive radio networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2190–2200, Aug 2014.

[8] X. Xing, T. Jing, H. Li, Y. Huo, X. Cheng, and T. Znati, "Optimal spectrum sensing interval in cognitive radio networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2408–2417, Sept 2014.

[9] B.-J. Kwak, N.-O. Song, and M. Miller, "Performance analysis of exponential backoff," *Networking, IEEE/ACM Transactions on*, vol. 13, no. 2, pp. 343–355, 2005.

[10] F. Tang, M. Guo, S. Guo, and C. Xu, "Mobility prediction based joint stable routing and channel assignment for mobile ad hoc cognitive networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, no. 99, pp. 1–1, 2013.

[11] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 114–128.