Delay-Minimized Routing in Mobile Cognitive Networks for Time-Critical Applications

Feilong Tang, *Senior Member, IEEE*, Can Tang, Yanqin Yang, Laurence T. Yang, *Senior Member, IEEE*, Tong Zhou, Jie Li, *Senior Member, IEEE*, and Minyi Guo, *Senior Member, IEEE*

***Abstract*—Cognitive radio significantly mitigates the spectrum scarcity for various applications built on wireless communication. Current techniques on mobile cognitive ad hoc networks (MCADNs), however, cannot be directly applied to time-critical applications due to channel interference, node mobility as well as unexpected primary user activities. In multichannel multiflow MCADNs, it becomes even worse because multiple links potentially interfere with each other. In this paper, we propose a delay-minimized routing (DMR) protocol for multichannel multiflow MCADNs. First, we formulate the DMR problem with the objective of delay minimization. Next, we propose a delay prediction model based on a conflict probability. Finally, we design the minimized path delay as a routing metric, and propose a heuristic joint routing and channel assignment algorithm to solve the DMR problem. Our DMR can find out the path with a minimal end-to-end (e2e) delay for time-critical data transmission. NS2-based simulation results demonstrate that our DMR protocol significantly outperforms related proposals in terms of average e2e delay, throughput, and packet loss rate.**

***Index Terms*—Channel assignment, delay prediction, mobile cognitive radio network, routing, signal collision.**

I. Introduction

WIRELESS communication is steadily increasing in various applications because it can offer several Advantages over traditional wired communication systems such as enhanced physical mobility and fewer infrastructure requirements [1]. Unfortunately, with the rapid growth of wireless applications, channel competition is becoming more and more serious due to the severe scarcity in the unlicensed spectrum. Cognitive radio is a promising technique to improve the efficiency of a licensed spectrum by the dynamic spectrum access. In mobile cognitive ad hoc networks (MCADNs), secondary users (SUs) frequently sense activities of primary users (PUs) and opportunistically access idle licensed channels of PUs [2].

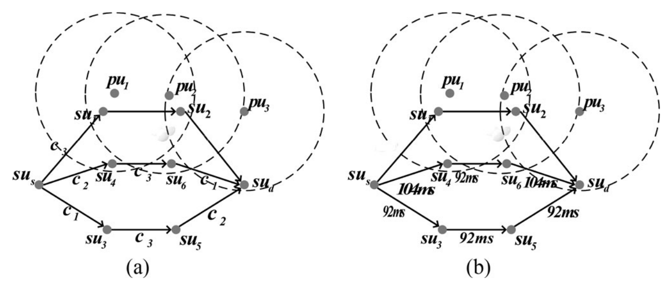


Fig. 1. Motivation scenario for our DMR protocol. (a) Joint routing and channel assignment. (b) Delay-minimized routing.

The uncertainty of PUs' activities, cochannel interference among SUs, and node mobility in MCADNs significantly affect the delay and reliability of time-critical applications. In multichannel multiflow MCADNs, it becomes even worse because multiple links potentially interfere with each other, which prevents MCADNs from real-time applications with a specified delay [3]. When a traffic accident occurs, a new navigation path should be sent to neighboring cars as soon as possible, for example, otherwise more cars will be blocked around the accident point.

To minimize the *end-to-end* (e2e) delay, we have to address two key issues in the design of routing protocols. First, routing and channel assignment should be jointly considered [4] because available channels of SUs are location- and time-dependent. As shown in Fig. 1, there are eight secondary users SU_s, SU_1–SU_6, and SU_d and three primary users PU_1, PU_2, and PU_3 in an MCADN. At this moment, 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 SU_s to SU_d. Suppose each SU can cover only its direct neighboring nodes, there are three path candidates for f_k: $P_1^{f_k} = \mathrm{SU}_s \rightarrow \mathrm{SU}_1 \rightarrow \mathrm{SU}_2 \rightarrow \mathrm{SU}_d$, $P_2^{f_k} = \mathrm{SU}_s \rightarrow \mathrm{SU}_4 \rightarrow$

Manuscript received November 22, 2015; revised April 4, 2016, June 21, 2016, and August 19, 2016; accepted September 2, 2016. Date of publication September 15, 2016; date of current version June 1, 2017. This work was supported in part by the National Natural Science Foundation of China projects under Grant 91438121, Grant 61373156, Grant 61672351, and Grant 61532013, in part by the National Basic Research Program (973 Program) under Grant 2015CB352403, and in part by the Huawei Technologies Co., Ltd., projects under Grant YB2015090040, Grant YBN2016090103 and Grant YB2015080089. Paper no. TII-15-1713. *(Corresponding author: F. Tang.)*

F. Tang and M. Guo are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: tang-fl@cs.sjtu.edu.cn; guo-my@cs.sjtu.edu.cn).

C. Tang is with the College of Business and Economics, The Australian National University, Canberra, A.C.T. 0200, Australia (e-mail: u5834561@anu.edu.au).

Y. Yang is with the Department of Computer Science and Technology, East China Normal University, Shanghai 200062, China (e-mail: yqyang@cs.ecnu.edu.cn).

L. T. Yang is with the Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada (e-mail: ltyang@gmail.com).

T. Zhou is with the School of Software, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zhoutong@sjtu.edu.cn).

J. Li is with the Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba 305-8571, Japan (e-mail: lijie@cs.tsukuba.ac.jp).

Digital Object Identifier 10.1109/TII.2016.2610408

$\mathrm{SU}_6 \rightarrow \mathrm{SU}_d$, and $P_3^{f_k} = \mathrm{SU}_s \rightarrow \mathrm{SU}_3 \rightarrow \mathrm{SU}_5 \rightarrow \mathrm{SU}_d$. Since SUs should not interfere with any PU, SU_2 has no available channel currently. As a result, $P_1^{f_k}$ cannot be selected for f_k. Instead, $P_2^{f_k}$ and $P_3^{f_k}$ can be respectively assigned the following interference-free channels such that $P_2^{f_k} = \mathrm{SU}_s \xrightarrow{c_2} \mathrm{SU}_4 \xrightarrow{c_3} \mathrm{SU}_6 \xrightarrow{c_1} \mathrm{SU}_d$ and $P_3^{f_k} = \mathrm{SU}_s \xrightarrow{c_1} \mathrm{SU}_3 \xrightarrow{c_3} \mathrm{SU}_5 \xrightarrow{c_2} \mathrm{SU}_d$, as shown in Fig. 1(a).

Second, it is also indispensable to set up a delay estimation model to capture a cochannel interference. 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 one is better in terms of the e2e delay. In Fig. 1(b), let the e2e delay of the two paths be $D_{P_2^{f_k}} = 104 + 92 + 104 = 300$ and $D_{P_3^{f_k}} = 92 + 92 + 92 = 276$ ms. In this case, we should select $P_3^{f_k}$ as the path of f_k, which integratedly considers the above two issues.

In recent years, the cross-layer design has been studied in the wireless networking for throughput optimization [5], [6], reliability guarantee [7], QoS-aware communication [8], and so on. Some researchers also proposed effective transmission time [9], [10] and spectrum switching delay [11] to control the e2e delay, based on cross-layer interactions. Unfortunately, these works cannot solve how to calculate the channel collision probability, while it is the key point of various delay models.

Based on the above analysis, in this paper, we investigate the *delay-minimized routing* (DMR) protocol for multichannel multiflow MCADNs, which can be used for real-time industrial applications, e.g., information services in vehicular ad hoc networks and wireless network-based indoor positioning system. Main contributions of this paper are summarized as follows:

1) We formulate the DMR problem to minimize the e2e delay in MCADNs for time-critical applications.
2) We propose an e2e delay prediction model, which consists of transmission time (TT) and media access time (MAT). Using the channel collision probability, we model expected transmission time (ETT) and expected media access time (EMAT) to quantitatively predict the e2e delay.
3) We propose a novel routing metric based on the ETT and EMAT to capture the channel interference.
4) Based on the proposed routing metric, we design a distributed DMR protocol for DMR and channel assignment in MCADNs.

The rest of the paper is organized as follows. In Section II, we briefly review related work. Section III describes the reference model of MCADNs and formulates the DMR problem. In Section IV, we present a delay prediction model based on the collision probability, and propose an e2e delay-based routing metric. Section V presents a new distributed DMR algorithm. In Section VI, we evaluate our DMR algorithm. Finally, Section VII concludes this paper.

II. Related Work

We review the related work about cross-layer design and e2e delay in wireless cognitive networks.

The cross-layer design typically requires a tight coupling between the routing and the spectrum management. Some works aimed to maximize network throughput [5], [6]. Pefkianakis *et al.* [5] pointed out that the network throughput is a function of time and spectrum, and estimated the spectrum utilization time to assign channels efficiently. Spectrum utility was proposed in [6], together with a new algorithm that is aimed to maximize throughput by jointly considering routing, spectrum allocation, and power control. Minimizing the interference between different sessions while providing throughput and reliability guarantee was investigated in [7], which jointly considered route selection and channel assignment.

Many classical e2e delay models take both routing and channel assignment into consideration. Chowdhury and Di Felice [12] used the hop number as the routing metric and modified the ad-hoc distance vector protocol for cognitive networks. However, these schemes cannot estimate the e2e path delay properly. A routing metric that takes spectrum switching delay and backoff delay into account was proposed in [11]. Moreover, the effective transmission time metric [9] captures the transmission delays of links. Spectrum-tree based on-demand routing protocol (STODRP) [10] uses a routing metric that combines transmission delay, channel switching delay, and protocol delay. In all these delay-oriented related proposals, the channel collision probability plays an important role to exactly estimate the delay. In cognitive networks, however, channel collision probability prediction is still a big challenge.

In television (TV) white space, unlicensed users need to periodically access a database to acquire the licensed spectrum usage. Calef and Cacciapuoti [13] proposed an optimal database access strategy to update the spectrum availability information, developed a stochastic analytical framework, and designed an algorithm able to efficiently compute the optimal strategy. Feng *et al.*[14] proposed a hybrid pricing framework for a TV white space database, through which the database operator optimally determines pricing parameters in terms of bandwidth reservation, registration fee, and query plans. Moreover, there are some excellent works on the mobility control; in [23], Yu *et al.* predict mobility through collecting GPS trajectory data, generating decision trees, and modeling the trees based on mobility regularities. Furthermore, they introduced the location popularity and location dependence to improve the prediction ability [24], [25].

III. System Model and Problem Formulation

A. System Model

An MCADN is modeled as an undirected graph $G(V, E)$, where V is the union of the SU set (V_S) and the PU set (V_P) such that $V = V_S \bigcup V_P$; E is the union of E_S (set of links among SUs) and E_P (set of links among PUs) such that $E = E_S \bigcup E_P$. R_T and R_I, respectively, denote transmission and interference ranges with $R_I = \beta \times R_T$ ($\beta \geq 2$).

Each $v \in V_S$ is equipped with q cognitive radios, which are able to detect available data channels [13], [15]. Each SU is also equipped with a traditional wireless interface, which forms a common control channel to transmit control messages, such as routing packets. In this paper, we assume that all nodes use an identical transmit power, and any two ways of a link be symmetrical. Let DC be a set of L orthogonal data channels

TABLE I
NOTATIONS

Notations	Meaning
$e_{v,u}$	A link from SUs v to u
$e^c_{v,u}$	A link-channel pair (i.e., $e_{v,u}$ is assigned a channel c)
$\mathrm{AC}_v(t)$	The set of available channels of an SU v at a slot t
f_k	The kth data flow $f_k=(s_k,d_k)$ $(1\le k\le K)$, where s_k and d_k are source and destination nodes of f_k
F	The set of all data flows
I_v	The set of PUs interfered by an SU v
$D^{f_k}_{e^c_{v,u}}$	Delay of f_k over a link $e_{v,u}$ and a channel c
T	The total transmission time of all data flows
r_v	Traffic generated by v in a slot
$r_{v,u}$	Traffic transmitting rate over $e_{v,u}$
N_v	Neighbor node set of v

in an MCADN, i.e., $\mathrm{DC} = \{c_1,\ldots,c_L\}$, and each channel $c_i \in \mathrm{DC}$ be assigned an identical bandwidth. As discussed by Kang *et al.* [16] and Fadda *et al.* [17], we assume to use TV white spaces, where it is realistic to sense the list of PU-free channels with ideal availability.

Two cognitive nodes v and u interfere with each other if they use the same channel and $\|u-v\|\le R_I$. Available channel set AC_v of an SU v changes with time and location due to the arbitrary appearance of PUs and node mobility. So, we divide continuous time as a series of discrete time slots and assume that AC_v and network topology keep fixed within any single time slot. Consequently, the available channel set of a link $e_{v,u} \in E_S$ in a slot t is $\mathrm{AC}_{v,u}(t)$=$\mathrm{AC}_v(t)\bigcap\mathrm{AC}_u(t)$.

B. Problem Formulation

The objective of this paper is to find paths with the minimal e2e delay in a multichannel multiflow MCADN. As shown in Fig. 1, route selection and channel assignment should be jointly considered. So we introduce the following three decision variables to indicate channel assignment [18], PU channel utilization, and path selection, respectively:

$$x^{f_k}_{c,e_{v,u}}(t) = \begin{cases} 1 & \text{if a channel c is assigned to } e_{v,u} \text{ for } f_k \\ & \text{at a time slot } t \\ 0 & \text{otherwise} \end{cases}$$

$$y^c_p(t) = \begin{cases} 1 & \text{if a PU } p \text{ is using its licensed channel } c \text{ at} \\ & \text{a time slot } t \\ 0 & \text{otherwise} \end{cases}$$

$$z^{f_k}_{e_{v,u}}(t) = \begin{cases} 1 & \text{if } f_k \text{ is being transmitted over } e_{v,u} \text{ at slot } t \\ 0 & \text{otherwise.} \end{cases}$$

Other important parameters are described in Table I. We formulate the DMR problem as follows.

Minimize

$$\sum_{f_k\in F}\sum_{e_{v,u}\in E_S} D^{f_k}_{e_{v,u}} z^{f_k}_{e_{v,u}}(t). \tag{1}$$

Subject to

$$\sum_{u\in N_v}(r_{u,v}-r_{v,u})+r_v=0,\ \ \forall v\in V_S; \tag{2}$$

$$\sum_{e_{s_k,u}\in E^o_{s_k}} z^{f_k}_{e_{s_k,u}}(t)=1,\ \ \forall f_k\in F; \tag{3}$$

$$\sum_{e_{w,u}\in E^o_w} z^{f_k}_{e_{w,u}}(t)=\sum_{e_{v,w}\in E^i_w} z^{f_k}_{e_{v,w}}(t),\ \ \forall w\in V_S\setminus\{s_k,d_k\},\ \ \forall f_k\in F; \tag{4}$$

$$x^{f_k}_{c,e_{v,u}}(t)+y^c_p(t)\le 1\ ,\forall e_{v,u}\in E_S,\ \ \forall p\in I_v\cup I_u\ \ \forall t\in[0,T] \tag{5}$$

$$x^{f_k}_{c,e_{v,u}}(t)\le z^{f_k}_{e_{v,u}}(t),\ \ \forall e_{v,u}\in E_S,\ \ \forall c\in\mathrm{DC},\ \ \forall t\in[0,T]; \tag{6}$$

$$\sum_{f_k\in F}\sum_{c\in\mathrm{AC}_{v,u}(t)} x^{f_k}_{c,e_{v,u}}(t)\le q,\ \ \forall e_{v,u}\in E_S,\ \ \forall c\in\mathrm{DC}\ \ \forall t\in[0,T]; \tag{7}$$

$$z^{f_k}_{e_{v,u}}(t)\in\{0,1\},\ \ \forall e_{v,u}\in E_S,\ \ \forall f_k\in F,\ \ \forall t\in[0,T]; \tag{8}$$

$$x^{f_k}_{c,e_{v,u}}(t)\in\{0,1\},\ \ \forall e_{v,u}\in E_S,\ \ \forall f_k\in F,\ \ \forall t\in[0,T]\ \ \forall c\in\mathrm{DC}; \tag{9}$$

$$y^c_p(t)\in\{0,1\},\ \ \forall p\in V_P,\ \ \forall t\in[0,T],\ \ \forall c\in\mathrm{DC}. \tag{10}$$

The objective of the DMR optimization in (1) is to minimize the total e2e delay for all the data flows. Constraint (2) guarantees a flow balance for each SU. Specifically, packets that any SU v sends out are equal to packets that v receives from its neighboring nodes plus traffic that v generates itself in a unit time. Constraints (3) and (4) guarantee that each data flow is transmitted along a single path from its source to destination. Specifically, a single path routing is ensured at any source node by constraint (3) and at any intermediate node by constraint (4). Here, $E^o_{s_k}$ is the set of links out of s_k. So, (3) means a flow f_k can only be transmitted over one link in $E^o_{s_k}$. We assume that a given node can be used to route packets for different data flows over different links, but each flow uses only one path from a source to a destination. Constraint (5) ensures that each SU will not impact any PU. Constraint (6) ensures that the channel must be allocated to the links selected in routes. Constraint (7) ensures that each node occupies at most q channels, where q is the number of radio equipped on a node. Finally, constraints (8)–(10) are binary value constraints ensuring that the decision variable can only take a value of 0 or 1. Note that t in the above formulation refers to a time slot for path setup and traffic passing; and a data session can be finished in multiple time slots.

The above-mentioned optimization problem is a nonlinear integer programming problem, in general it is NP-hard. In this paper, we will propose a heuristic DMR algorithm in Section V.

IV. Delay Prediction

The basic principle of our routing protocol is to minimize the e2e delay. In this section, we present how to predict the delay of a link-channel $e^c_{v,u}$, and then formulate the e2e delay of a path.

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

$$D^{f_k}_{e^c_{v,u}} = \mathrm{EMAT}^c_{v,u} + \mathrm{ETT}^c_{v,u} \tag{11}$$

where $\mathrm{EMAT}^c_{v,u}$ and $\mathrm{ETT}^c_{v,u}$ are EMAT and expected transmission time of $e^c_{v,u}$, respectively. So, we have the link delay $D^{f_k}_{e_{v,u}} = \min\{D^{f_k}_{e^c_{v,u}} | \forall c \in \mathrm{AC}_{v,u}(t)\}$.

A. Expected Transmission Time

In MCADNs, channel interference among links will cause a packet retransmission, which increases the link delay and furthers the e2e path delay and decreases the network throughput. Since it is impossible to get accurate MAT and TT in dynamical MCADNs [19], we use the expected transmission time to measure the packet transmission delay.

Assume that $P^c_{v,u}$ be the collision probability of a link channel $e^c_{v,u}$ and $\mathrm{ETX}^c_{v,u}$ be the expected times that a given packet is transmitted from v to u on a channel c. We can get $\mathrm{ETT}^c_{v,u}$ and $\mathrm{ETX}^c_{v,u}$ using (12) and (13), respectively

$$\mathrm{ETT}^c_{v,u} = \mathrm{ETX}^c_{v,u} \times \left(T^{c,\mathrm{DATA}}_{v,u} + T^{c,\mathrm{ACK}}_{u,v}\right) \tag{12}$$

$$\begin{aligned}\mathrm{ETX}^c_{v,u} &= 1 \times (1 - P^c_{v,u}) + 2 \times (1 - P^c_{v,u}) \times P^c_{v,u} \\ &\quad + \cdots + n \times (1 - P^c_{v,u}) \times (P^c_{v,u})^{n-1} + \cdots \\ &= \frac{1}{1 - P^c_{v,u}}\end{aligned} \tag{13}$$

where $T^{c,\mathrm{ACK}}_{u,v}$ is the time when v receives an ACK message from u, and $T^{c,\mathrm{DATA}}_{v,u}$ is the time of transmitting a data packet over a link channel $e^c_{v,u}$,which can be obtained as follows:

$$T^{c,\mathrm{DATA}}_{v,u} = \frac{S_{\mathrm{packet}}}{B^c} \tag{14}$$

where S_{packet} is the size of a data packet and B^c is the bandwidth of channel c on the link $e_{v,u}$. $T^{c,\mathrm{ACK}}_{u,v}$ can be calculated similarly.

As shown in (13), $\mathrm{ETX}^c_{v,u}$ closely depends on the collision probability $P^c_{v,u}$ of $e^c_{v,u}$. In the following, we present how to calculate the collision probability $P^c_{u,v}$ of a link channel $e^c_{v,u}$.

In Fig. 2, v is in the transmission range of u, and M_1, M_2, M_3, and N_3 are in the interference range of u. In MCADNs, channel collision falls into the following three categories:

1) *Data packet collision.* M_1 sends data packets to N_1, meanwhile v sends data packets to u. Data packets from M_1 and v cause collision at u, with the probability $P1^c_{v,u}$.
2) *ACK collision.* N_2 sends data packets to M_2, and M_2 responds ACKs 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.* M_3 sends data packets to N_3, and N_3 sends ACK packets to M_3. Meanwhile, v sends data packets to u. So, ACK packets from N_3 and data packets from M_3 cause collision with data packets from v. The collision probability is denoted as $P3^c_{v,u}$.

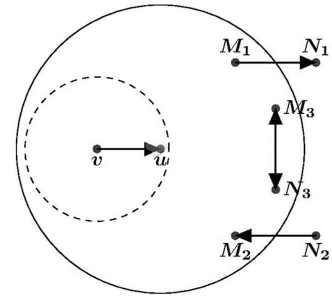


Fig. 2. Three categories of collisions.

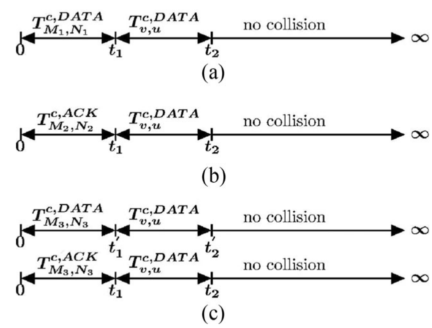


Fig. 3. Collision probability in the three categories of collisions. (a) Data packet collision. (b) ACK collision. (c) Data and ACK collisions.

Let u be interfered by a link $e_{m,n}$. Collision categories depend on the distance between u and $\omega \in \{m, n\}$, and the type of the packet that ω is sending. If both m and n are in the interference range of u, the collision is of the third category. If only m is in the interference range of u, the collision will be of the first category when m is sending a data packet or the second category when m is sending an ACK.

Assume that packets be sent in the Poisson distribution. Specifically, a link channel $e^c_{m,n}$ sends data packets at a rate $\lambda^c_{m,n}$, and sends ACK packets at a rate $\gamma^c_{m,n}$. We can get the probability $P^c_{m,n}(X = 0)$ of the link channel $e^c_{m,n}$, which refers to the probability of $e^c_{m,n}$ sending zero packet in a unit time, in the formulas (15) and (16). As a result, the probability $P^c_{m,n}(X = 0, t = T)$ that $e^c_{m,n}$ sends zero packet during T slots can be formulated in formulas (17) and (18):

$$P^{c,\mathrm{DATA}}_{m,n}(X = 0) = \exp\{-\lambda^c_{m,n}\} \tag{15}$$

$$P^{c,\mathrm{ACK}}_{m,n}(X = 0) = \exp\{-\gamma^c_{m,n}\} \tag{16}$$

$$P^{c,\mathrm{DATA}}_{m,n}(X = 0, t = T) = (\exp\{-\lambda^c_{m,n}\})^T \tag{17}$$

$$P^{c,\mathrm{ACK}}_{m,n}(X = 0, t = T) = (\exp\{-\gamma^c_{m,n}\})^T. \tag{18}$$

Without loss of generality, we suppose that v starts sending data packets to u at a time slot t_1, as shown in Fig. 3(a). If M_1

sends data packets to N_1 during the 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. So, we can calculate the $P1^c_{v,u}$ as follows:

$$\begin{aligned} P1^c_{v,u} &= 1 - P^{c,\text{DATA}}_{M_1,N_1}\left(X = 0, t = T^{c,\text{DATA}}_{M_1,N_1} + T^{c,\text{DATA}}_{v,u}\right) \\ &= 1 - \exp\left\{-\lambda^c_{M_1,N_1}\left(T^{c,\text{DATA}}_{M_1,N_1} + T^{c,\text{DATA}}_{v,u}\right)\right\}. \end{aligned} \tag{19}$$

Similarly, we can get $P2^c_{v,u}$ and $P3^c_{v,u}$ with formulas (20) and (21), as shown in Fig. 3(b) and (c), respectively:

$$\begin{aligned} P2^c_{v,u} &= 1 - P^{c,\text{ACK}}_{N_2,M_2}\left(X = 0, t = T^{c,\text{ACK}}_{N_2,M_2} + T^{c,\text{DATA}}_{v,u}\right) \\ &= 1 - \exp\left\{-\gamma^c_{N_2,M_2}\left(T^{c,\text{ACK}}_{N_2,M_2} + T^{c,\text{DATA}}_{v,u}\right)\right. \end{aligned} \tag{20}$$

$$\begin{aligned} P3^c_{v,u} &= 1 - P^{c,\text{DATA}}_{M_3,N_3}\left(X = 0, t = \left(T^{c,\text{DATA}}_{M_3,N_3} + T^{c,\text{DATA}}_{v,u}\right)\right) \\ &\quad \times P^{c,\text{ACK}}_{N_3,M_3}\left(X = 0, t = \left(T^{c,\text{ACK}}_{N_3,M_3} + T^{c,\text{DATA}}_{v,u}\right)\right) \\ &= 1 - \exp\left\{-\left\{\lambda^c_{M_3,N_3}\left(T^{c,\text{DATA}}_{M_3,N_3} + T^{c,\text{DATA}}_{v,u}\right)\right.\right. \\ &\quad \left.\left. + \gamma^c_{N_3,M_3}\left(T^{c,\text{ACK}}_{N_3,M_3} + T^{c,\text{DATA}}_{v,u}\right)\right\}\right\}. \end{aligned} \tag{21}$$

Assume that channel collisions be independent of each other so that we can get $P^c_{v,u}$ using the following:

$$\begin{aligned} P^c_{v,u} = 1 - &\prod_{e^c_{m,n} \in I^1(e^c_{v,u})} \exp\left\{-\lambda^c_{m,n}\left(T^{c,\text{DATA}}_{m,n} + T^{c,\text{DATA}}_{v,u}\right)\right\} \\ &\times \prod_{e^c_{m,n} \in I^2(e^c_{v,u})} \exp\left\{-\gamma^c_{m,n}\left(T^{c,\text{ACK}}_{m,n} + T^{c,\text{DATA}}_{v,u}\right)\right\} \\ &\times \prod_{e^c_{m,n} \in I^3(e^c_{v,u})} \exp\left\{-\left\{\lambda^c_{m,n}\left(T^{c,\text{DATA}}_{m,n} + T^{c,\text{DATA}}_{v,u}\right)\right.\right. \\ &\left.\left. + \gamma^c_{m,n}\left(T^{c,\text{ACK}}_{m,n} + T^{c,\text{DATA}}_{v,u}\right)\right\}\right\} \\ \approx &\sum_{e^c_{m,n} \in I^1(e^c_{v,u})} \lambda^c_{m,n}\left(T^{c,\text{DATA}}_{m,n} + T^{c,\text{DATA}}_{v,u}\right) \\ &+ \sum_{e^c_{m,n} \in I^2(e^c_{v,u})} \gamma^c_{m,n}\left(T^{c,\text{ACK}}_{m,n} + T^{c,\text{DATA}}_{v,u}\right) \\ &+ \sum_{e^c_{m,n} \in I^3(e^c_{v,u})} \lambda^c_{m,n}\left(T^{c,\text{DATA}}_{m,n} + T^{c,\text{DATA}}_{v,u}\right) \\ &+ \gamma^c_{m,n}\left(T^{c,\text{ACK}}_{m,n} + T^{c,\text{DATA}}_{v,u}\right) \end{aligned} \tag{22}$$

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

B. Expected Media Access Time

In time-slot-based 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 are randomly selected from $\{0, 1, \ldots, W_0 - 1\}$, 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 EMAT. Based on [20], we use the following formula to calculate the EMAT:

$$\overline{D_{\sum}}^c_{v,u} = \frac{1}{2}\left(\frac{1}{1 - P^c_{v,u}} + \frac{W_0}{1 - r \times P^c_{v,u}}\right) - 1 \tag{23}$$

$$\text{EMAT}^c_{v,u} = \overline{D_{\sum}}^c_{v,u} \times \text{slot} \tag{24}$$

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 $e^c_{v,u}$; slot is a waiting time unit; and r is the expand factor of W_0. Note that a similar procedure is adopted for the ACKs.

C. Expected e2e Path Delay

According to Sections IV-A and IV-B, we can calculate the link delay in formula (11) in terms of the collision probability. Now, it is easy to calculate the e2e path delay $D^{f_k}_{\text{path}}$ for a flow f_k by accumulating the delay of links involved in the path, formulated as

$$D^{f_k}_{\text{path}} = \sum_{e_{v,u} \in f_k} D^{f_k}_{e_{v,u}}. \tag{25}$$

V. Joint Routing and Channel Assignment Based on the Minimal Path Delay

In this section, we present our DMR protocol, using the e2e path delay $D^{f_k}_{\text{path}}$ in (25) as the routing metric. In our DMR, each cognitive node makes decisions on route selection and channel assignment based on its local information.

A. Link Stability Prediction

Besides the signal interference, in MCADNs, the duration of a link $e_{v,u}$ also significantly suffers from the relative movement of v and u, and PU activation. So we need to predict *maximal lifetime* $\text{MLT}^c_{v,u}$ of a link channel $e^c_{v,u}$ due to node mobility and PU activities. If $\text{MLT}^c_{v,u} < D^{f_k}_{e^c_{v,u}}$, $e_{v,u}$ should not be involved in the route.

We first assume that the node mobility follows the random waypoint model. When the distance $d_{v,u}(t)$ between v and u is equal to R_T, t reaches the mobility duration $T_{v,u}$. So, we can solve $T_{v,u}$ through setting $d_{v,u}(t) = R_T$. Similarly, let $d_{v,m}(t) = R_I$, we can easily make sure the longest duration T^c_{v,PU_m} that measures how long node v using channel c will interfere with PU_m, where PU_m is using its channel c. Note that we suppose different PUs use different licensed channels in an MCADN. If $T_{v,u} \le T^c_{v,\text{PU}_m}$, node v will not affect any PU because $e_{v,u}$ has lost before it interferes with PU_m. Then, we can get $\text{MLT}^c_{v,u}$ as follows:

$$\text{MLT}^c_{v,u} = \min\{T_{v,u}, T^c_{v,\text{PU}_m}, T^c_{u,\text{PU}_m}\}. \tag{26}$$

Next, we introduce a probability to capture random changes of the nodes' velocity [21]. Let $P(\text{MLT}^c_{v,u})$ refer to the probability

TABLE II
RREQ MESSAGE FORMAT

Type	Reserved	Hop count	RREQ id
Destination address		Originator address	
Destination sequence number		Originator sequence number	
Total delay			
channelClsPro[0]		channelClsPro[1]	
...			
channelClsPro[10]		channelClsPro[11]	
Node address	Link channel	Link delay	

that link channel $e^c_{u,v}$ is available by the end of $\mathrm{MLT}^c_{v,u}$. It can be estimated as

$$P(\mathrm{MLT}^c_{v,u}) \approx e^{-\lambda \mathrm{MLT}^c_{v,u} \times e^{-\lambda\tau}} + \zeta(1 - e^{-\lambda \mathrm{MLT}^c_{v,u}}) \tag{27}$$

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

$\mathrm{MLT}^c_{v,u}$ is the maximal time period that a currently available link can keep if no change in velocities occurs. $P(\mathrm{MLT}^c_{v,u})$ captures possible changes in velocities that may occur during the period $\mathrm{MLT}^c_{v,u}$. So, we predict the expected maximal lifetime $\mathrm{EMLT}^c_{v,u}$ of $e^c_{v,u}$ with the formula (28), which considers random changes of nodes' speeds and directions:

$$\mathrm{EMLT}^c_{v,u} = \mathrm{MLT}^c_{u,u} \times P(\mathrm{MLT}^c_{u,u}). \tag{28}$$

B. DMR Protocol

In the DMR 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 as 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 exchange the channel information (e.g., rate of sending data packets and ACK packets on the channel) and calculate the collision probability using formulas (22) and (29). We can get the collision probability P^c_v of v on each available channel c using the following:

$$P^c_v = 1 - \prod_{u \in N_v} (1 - P^c_{v,u}) \tag{29}$$

where N_v denotes the set of neighboring nodes of v.

1) Joint Path Selection and Channel Assignment: It consists of two phases: *routing request* (RREQ) described in Algorithm 1 and *route setup* in Algorithm 2. The latter inserts a path found in Algorithm 1 into routing tables of involved nodes.

Initially, a source node s_k broadcasts an RREQ packet with the format presented in Table II. On receiving an RREQ packet, a relay node v performs the DMR algorithm, which will be run hop by hop until a destination d_k is reached in a distributed fashion. Algorithm 1 describes the process of route discovery, where rq and rp are RREQ and routing reply (RREP) packets, respectively; p.delay refers to the total transmission time from s_k to the relay node v, i.e., "total delay" in RREQ and RREP packets; and $\mathrm{AC}_{v,u}(t)$ is the available channel set of $e_{v,u}$ in a slot t.

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 $= \infty$
minChannel $= 0$
for ($\forall c \in \mathrm{AC}_{v,u}(t)$) **do**
  calculate $D^{f_k}_{e^c_{u,v}}$ using (11)
  calculate $\mathrm{EMLT}^c_{u,v}$ using (28)
  if ($\mathrm{EMLT}^c_{u,v} > p.\mathrm{delay} + D^{f_k}_{e^c_{u,v}}$)$\wedge$ ($D^{f_k}_{e^c_{u,v}} \le$ minDelay) **then**
    minDelay $= D^{f_k}_{e^c_{u,v}}$
    minChannel $= c$
  end if
end for
p.delay += minDelay
get reverse route $\mathrm{rt_{src}}$ from v's routing table
get route $\mathrm{rt_{dest}}$ from v's routing table
if ($\mathrm{rt_{src}}$ = NULL) $\vee$ ($\mathrm{rt_{src}}$.delay $\ge p$.delay) **then**
  update $\mathrm{rt_{src}}$
else
  return
end if
if (v = p.rp_dest) $\vee$ ($\mathrm{rt_{dest}} \ne$ NULL) **then**
  replay an RREP packet with $\mathrm{rt_{src}}$
else
  broadcast p
end if

During a route setup, cognitive nodes use the DMR algorithm to carry out its local optimal choice hop by hop toward a destination node. As shown in Algorithm 1, when a node v receives an 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 information in the RREQ packet to calculate the link delay $D^{f_k}_{e^c_{u,v}}$ of $e_{u,v}$ for each available channel c. If $\mathrm{EMLT}^c_{u,v} > p.\mathrm{delay} + D^{f_k}_{e^c_{u,v}}$, it means that the link channel $e^c_{u,v}$ can be used in the route. Then, v assigns a channel with the minimal link delay (minDelay) for $e_{u,v}$.
3) v accumulates the minDelay to the delay p.delay. If the expected delay is smaller than the previous delay in the route table, v will replace the old reverse route with the new reverse route from s_k to v.
4) If v has a route to d_k in its route table or v is d_k, v makes an RREP packet and sends it along with a current reverse route. Otherwise, it will broadcast RREQ. The format of the RREP packet is shown in Table III.

Algorithm 1 chooses the best link channel $e^c_{u,v}$ hop by hop. Finally, we can get a selected path with minimal e2e delay and assigned channels from s_k to d_k, summarized in Theorem 1.

TABLE III
RREP MESSAGE FORMAT

Type	Reserved	Hop count
RREQ id		
Destination address	Destination sequence number	
Originator address	Originator sequence number	
Total delay		

Theorem 1: Algorithm 1 can get a path $P_{f_k}^{\min}$ with the minimal e2e delay for any flow f_k.

Proof: We use a contradiction to proof Theorem 1.

Let there be a path $P_{f_k}^{\min}$ with a minimal e2e delay that cannot be found out by Algorithm 1. It could be weeded out only in steps 2 and 21. Let delay(P) denote the delay of a path P.

1) Let $P_{f_k}^{\min}$ be weeded out in step 2 when v receives RREQ from u (v and u are intermediate nodes of $P_{f_k}^{\min}$). So, we can deduce that the path $P_{f_k}^{\min} = s_k \to \cdots \to u \to v \to \cdots \to d_k$ and v is s_k. Then, we can find a new path $P^{'} = v \to \cdots \to d_k$ that is just a subpath of $P_{f_k}^{\min}$. This means that the e2e delay of path P is smaller than that of $P_{f_k}^{\min}$. This conclusion contradicts with the assumption.
2) If $P_{f_k}^{\min}$ is weeded out in step 21 when v receive an RREQ packet from u, we have the path $P_{f_k}^{\min} = s_k \to \cdots \to u \to v \to \cdots \to d_k = P_{\text{sub1}} + P_{\text{sub2}}$, where $P_{\text{sub1}} = s_k \to \cdots \to u \to v$ and $P_{\text{sub2}} = v \to \cdots \to d_k$. Since $P_{f_k}^{\min}$ was weeded out at line 21, there is another path $P^{'}$ from s_k to node v with $\text{delay}(P^{'}) < \text{delay}(P_{\text{sub1}})$. So we can deduce that there is a new path $P^{''} = P^{'} + P_{\text{sub2}}$. And $\text{delay}(P^{''}) = \text{delay}(P^{'}) + \text{delay}(P_{\text{sub2}}) < \text{delay}(P_{\text{sub1}}) + \text{delay}(P_{\text{sub2}}) = \text{delay}(P_{f_k}^{\min})$. This is also contradicted with the assumption that the path $P_{f_k}^{\min}$ has the minimum delay.

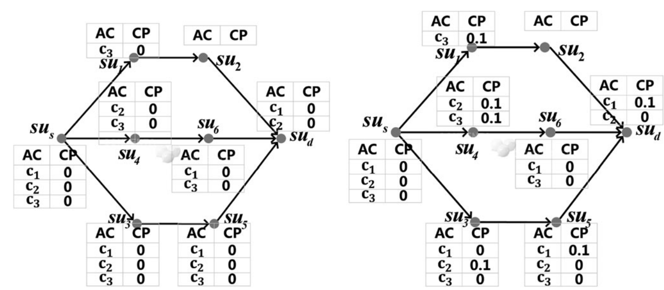
Moreover, steps 7–14 in Algorithm 1 guarantee that the selected relay $v \in N_u$ enables the link channel $e_{u,v}^c$ to have the minimal link delay $D_{e_{u,v}^c}^{f_k}$ among all neighbors of u. And steps 18–22 update routing tables of all relays with the route that has the minimal path delay. Consequently, these steps enable Algorithm 1 to preserve the relay-beneficial condition, the strict preference preservation, and the relay order optimality. So, we can conclude that Algorithm 1 can choose the path $P_{f_k}^{\min}$ with a minimal e2e delay for any flow f_k. Here, the minimal e2e delay refers to the solution solved by Algorithms 1 and 2. In fact, it is a suboptimal solution. ∎

2) Route Setup: When a node u receives an 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 the RREP packet is lower than that in rt_{dest}, u updates its route table.

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) ∧ (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
```



(a) (b)

Fig. 4. Channel collision tables of SUs. (a) Initially. (b) During path setup.

3) If u is not the source node, it updates its route to rt_{dest} in a route table based on the RREP packet. u forwards this RREP packet to the next hop using rt_{src}.

C. Case Study

We use the scenario described in Section I to illustrate our DMR protocol.

At the beginning of the DMR, each cognitive node in the network has no route record in its route table, as shown in Fig. 5(a). And the collision probability of each available channel is 0, as shown in Fig. 4(a). When SU_s attempts to send packets to SU_d, each node's channel collision probability is shown in Fig. 4(b).

When the RREQ reaches SU_1, SU_1 calculates the minimal link delay = 104 ms from SU_s to itself [according to the channel collision probability table shown in Fig. 4(b)]. SU_1 first records the reverse route to SU_s in its route table, as shown in Fig. 5(b). Then, it accumulates the RREQ delay, and records its channel collision probability table and link-channel selection $\text{SU}_s \xrightarrow{c_3} \text{SU}_1$ into RREQ. Finally, SU_1 broadcasts the RREQ. When RREQ reaches SU_3–SU_6 they just carry out the similar work as SU_1. SU_2 does not broadcast RREQ because it has no any available channel. We suppose that the RREQ packet on $P_2^{f_k} = \text{SU}_s \xrightarrow{c_2} \text{SU}_4 \xrightarrow{c_3} \text{SU}_6 \xrightarrow{c_1} \text{SU}_d$ first reaches SU_d. SU_d just carries out the similar work as SU_1, then makes RREP packet for the path $P_2^{f_k}$ and sends

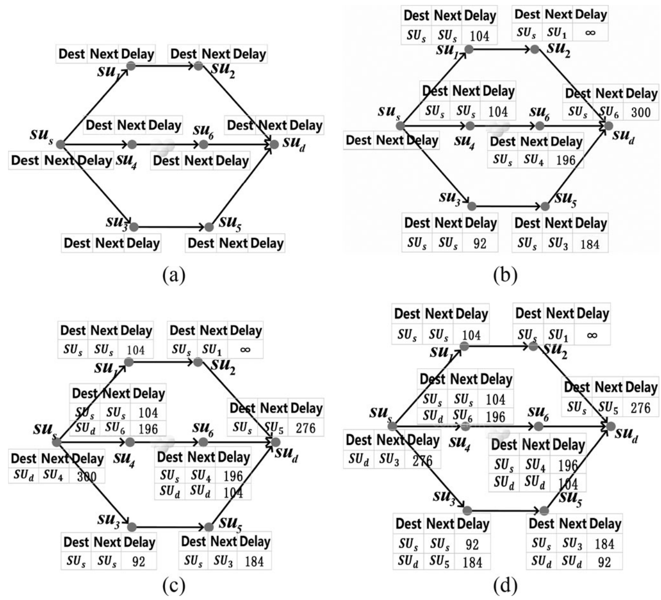


Fig. 5. Route tables of SUs. (a) Initially. (b) $P_2^{f_k}$ RREQ reached SU$_d$. (c) $P_3^{f_k}$ RREQ reached SU$_d$ and $P_2^{f_k}$ RREP reached SU$_s$. (d) $P_3^{f_k}$ RREP reached SU$_s$.

it using $P_2^{f_k}$. At the same time, another RREQ packet on $P_3^{f_k} = \mathrm{SU}_s \xrightarrow{c_1} \mathrm{SU}_3 \xrightarrow{c_3} \mathrm{SU}_5 \xrightarrow{c_2} \mathrm{SU}_d$ reaches SU$_5$. All nodes' route tables are shown in Fig. 5(b).

When the second RREQ packet reaches SU$_d$, it also calculates the link delay $\mathrm{Delay}_{e_{5,d}}^{c_1} = 92$ ms, and accumulates delay in the RREQ packet. Then, SU$_d$ realizes that the e2e delay of $P_3^{f_k} = \mathrm{SU}_s \rightarrow \mathrm{SU}_3 \rightarrow \mathrm{SU}_5 \rightarrow \mathrm{SU}_d$ is smaller than that of $P_2^{f_k}$. So, SU$_d$ replaces $P_2^{f_k}$ with $P_3^{f_k}$ in its route table, as shown in Fig. 5(c). By the way, when the RREP packet of $P_2^{f_k}$ reaches SU$_4$, SU$_6$, and SU$_s$, they record the path $P_2^{f_k}$ to node SU$_d$ in their route tables. Note that SU$_s$ records $P_2^{f_k}$ in its route table because $P_2^{f_k}$ RREP reaches SU$_s$ earlier than $P_3^{f_k}$, shown in Fig. 5(c).

Finally, when the $P_3^{f_k}$ RREP reaches SU$_s$, SU$_s$ also realizes that the e2e delay of $P_3^{f_k}$ is smaller than that of $P_2^{f_k}$, and accordingly replaces $P_2^{f_k}$ with $P_3^{f_k}$ in its route table, as shown in Fig. 5(d). At this moment, DMR ends and SU$_s$ finds a path to SU$_d$ with the minimal e2e delay.

D. Local Route Repair

When a route fails at the link channel $e_{v,u}^c$, v broadcasts a route error packet (includes the disabled link channel $e_{v,u}^c$, destination d_k, and collision probability of available channels of v) to neighbors. When a neighbor $n_v \in N_v$ receives the route error packet, n_v performs a local route repair as follows:

1) n_v checks its route table. If there is a route 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 the route table of n_v, n_v replies with an error response packet (include a minimal link-channel delay $D_{e_{v,n_v}^c}^{f_k}$, path $P_{n_v,d_k}^{f_k}$ from n_v to destination node, and the e2e delay of $P_{n_v,d_k}^{f_k}$) to node v.
3) Whenever v receives an error response packet, if there is no usable route to destination or new e2e delay is better than old one, v updates its route table and sends this route to the source node along with the reverse route.
4) If v does not 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 the route error packet reaches the source node s_k, s_k will make a route request packet once more and use the DMR protocol to get the minimal e2e delay path.

TABLE IV
SYSTEM PARAMETERS

Parameters	Value
Number of PUs	2
Number of SUs	60
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	50μ s
Maximal speed	10 m/s

VI. SIMULATION AND EVALUATION

We developed a simulation system, which was built on the network simulator V2 (NS2) with multiradio multichannel extensions, to evaluate our DMR protocol. First, we describe the simulation system setting. We, then, comprehensively evaluate our DMR by comparing it with the related and the most recent proposals in terms of various performance metrics.

A. Simulation System Setting

In our system, PUs and SUs are randomly deployed in an area of 2500 m $\times$ 2500 m. Both SUs and PUs move at a speed randomly distributed in $[0, V_{\max}]$. Channels comply with the two-way Rayleigh model. Signal propagation was set as a two-ray ground reflection model. SUs and PUs were set at the same transmission radius and interference range. Each PU was assigned a fixed data channel, which was randomly used.

Each SU has multiple available data channels. In the experiments, we tested different performance metrics for 1000 s. Each flow is generated through an NS2-based FTP data generator. A slot is one basic waiting time unit in (24) and we use the default setting slot $= 50\ \mu$ s. The simulated raw bit rate is set as 1Mbps. More parameters are listed in Table IV.

B. Performance Evaluation

We evaluate our DMR protocol by comparing it with the well-known routing protocols for cognitive networks.

1) STODRP [10]. STODRP uses the ETT, protocol delay, and channel access delay as a routing metric. It builds a spectrum tree ST^c for each available channel c and adds

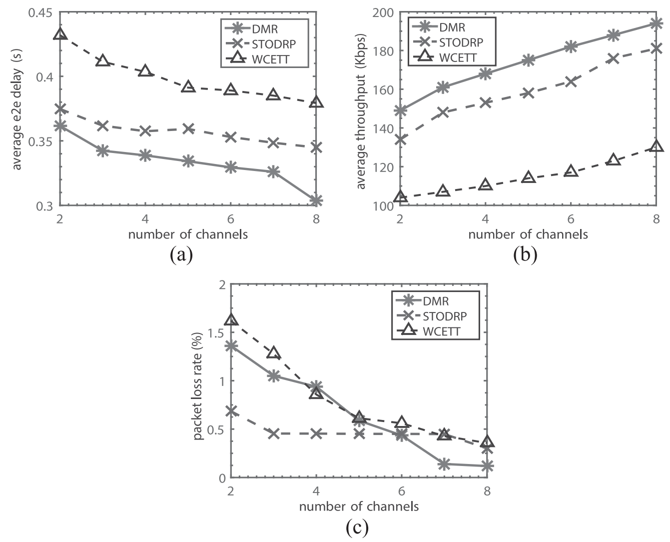


Fig. 6. Performance with the number of channels. (a) Average e2e delay. (b) Average throughput. (c) Packet loss rate.

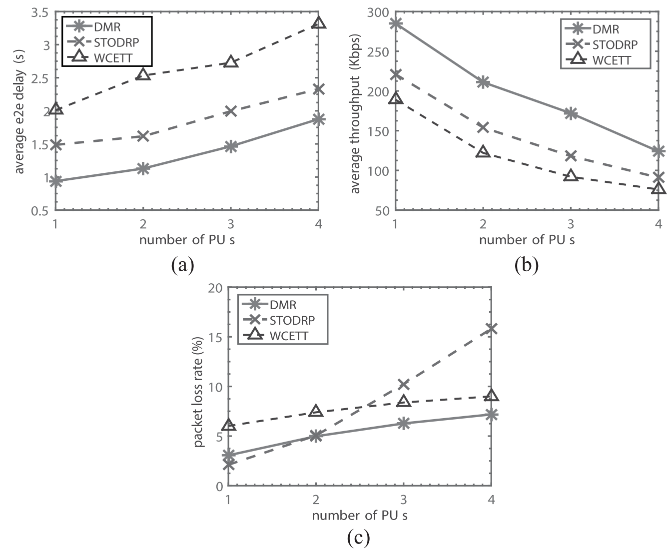


Fig. 7. Performance with the number of PUs. (a) Average e2e delay. (b) Average throughput. (c) Packet loss rate.

the nodes with the available channel c into ST^c. Whenever a node s_k wants to start a flow, it first sends a route request to a root node r_c. If the destination node d_k is in the same spectrum tree, s_k sends RREQ in the local spectrum tree. Otherwise, r_c chooses an interspectrum node v with the shortest queuing size to relay packets from s_k.

2) WCETT (*weighted cumulative expected transmission time*) [22]. It is also a delay-oriented routing protocol. WCETT establishes routes such as Ad hoc On-demand Distance Vector Routing (AODV), and it considers ETT and channel interference only along an intraflow. However, it does not consider channel interference among interflows. So, channel interference among interflows increases sharply in WCETT as the number of flows increases. Besides, WCETT does not propose an effective channel collision model to estimate ETT.

In the following, we evaluate how various performance metrics of the three protocols change with the number of channels, PUs, SUs, flows, speed, and transmission rate.

1) Performance With Channels: The number of channels significantly affects the e2e performance. As shown in Fig. 6(a) and (b), as more channels are added into the network, e2e delay and average throughput, respectively, decrease and increase step by step in the three proposals. But, our DMR always exhibits the shortest e2e delay and the highest throughput among them. As the number of channels increases, for example, the e2e delay in STODRP is around 12% longer than that in our DMR [see Fig. 6(a)] and throughput in our DMR is around 13% higher than that in STODRP on an average [see Fig. 6(b)]. The reason is that our DMR always selects a route with the shortest e2e delay, and assigns conflict-free channels for data flows. Instead, WCETT does not consider how to avoid interference among interflows, which increases the e2e delay and decreases the throughput. In STODRP, the more the channels added, the more spectrum trees need to be built and updated at each root node, which significantly impact e2e delay and throughput.

From Fig. 6(c), we can find that more channels can reduce the packet loss rate in the three schemes. STODRP records information of each node so that it can decrease the packet loss rate by choosing the node with the shortest queue to transmit packets. When the number of channels is up to 6, however, our DMR will outperform both STODRP and WCETT in terms of the packet loss rate because spectrum trees impose too high building and updating cost in STODRP.

2) Performance With PU Number: In cognitive networks, more PUs will bring higher uncertainty of links among SUs because SUs cannot affect communications of any PU. As shown in Fig. 7, as the number of PUs increases, both e2e delay and loss rate increase, and throughput decreases in the three proposals. However, our DMR always outperforms STODRP and WCETT. From Fig. 7, for example, the e2e delay in STODRP is around 35% longer than that in our DMR and the throughput in our DMR is around 37.5% higher than that in STODRP on the average as the number of PUs increases. The reason is that maintaining spectrum trees in STODRP will become more and more complex with increase in PU number because STODRP tries to use spectrum availability to partition cognitive nodes into different spectrum trees. So the e2e delay, throughput, and packet loss rate in STODRP become worse than those in DMR, as shown in Fig. 7(a)–(c). Similarly, the more the PUs added in the network, the more seriously the performance in WCETT will be affected. The reason is that WCETT focuses on how to explicitly account for the interference among links that use the same channel [22] rather than how to assign conflict-free channels.

3) Performance With SU Number: We used different number of SUs to test the three proposals. As shown in Fig. 8, average delay, throughput, and packet loss rate all increase as SUs increase. But our DMR can always perform the lowest delay and packet loss rate and the highest throughput among the three proposals in all scenarios. Note that we increased two more flows whenever we increased ten SUs.

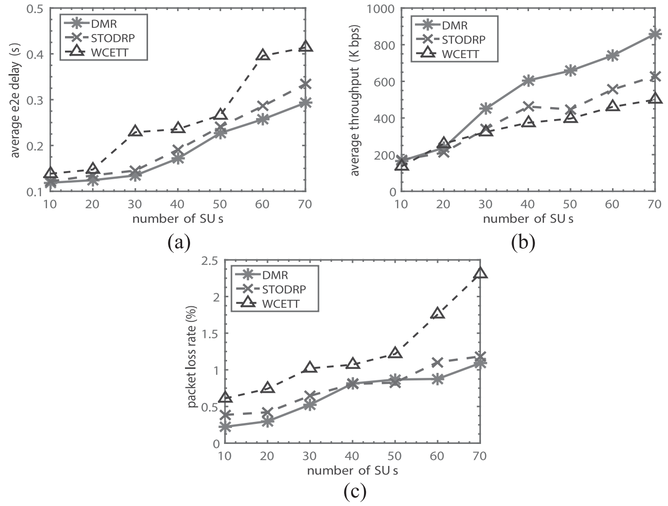


Fig. 8. Performance with the number of SUs. (a) Average e2e delay. (b) Average throughput. (c) Packet loss rate.

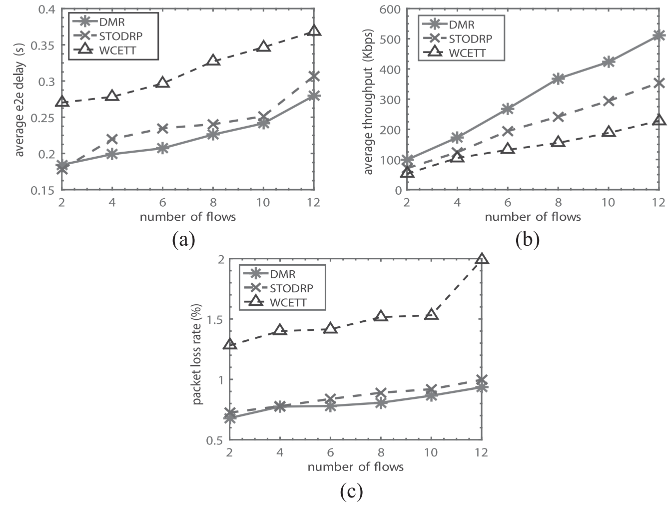


Fig. 9. Performance with the number of data flows. (a) Average e2e delay. (b) Average throughput. (c) Packet loss rate.

In STODRP, more secondary nodes mean a more complex spectrum tree because each root node records more information and the process of a spectrum route request between different spectrum trees causes high cost. So, STODRP exhibits a lower throughput than DMR. WCETT just uses the calculated ETT as its route metric so that it cannot estimate the e2e delay accurately. So, the e2e delay in WCETT is longer than that in DMR and STODRP. Moreover, WCETT does not consider how to avoid channel interference between interflows so that more data flow will result in higher packet loss rate. Note that the e2e delay increases with the number of SUs because hops will accordingly increase in a pair of given source and destination, which results in more serious signal interference among these nodes involved in routes. On the other hand, more dense nodes enable the route setup more quickly so that the average throughput also accordingly increases.

4) Performance With Data Flows: In a given network, more flows will potentially cause more signal conflict. From Fig. 9, we can find that as data flows increase, average e2e delay, throughput, and packet loss rate increase in all proposals. Our DMR, however, always exhibit better performance than STODRP and WCETT. The reason is that both DMR and STODRP consider channel collision among different flows but WCETT does not take it into account. So, WCETT exhibits higher e2e delay and packet loss rate, and lower throughput than other two proposals due to serious channel interference.

After analyzing the results in Fig. 9, we can conclude that building and updating spectrum trees in STODRP introduce high cost and our DMR has higher performance. The reason is that as flows increase, channel competition among these flows become more and more serious so that available channels of each node change sharply. So, building and updating spectrum trees in STODRP inevitably become more frequent. Note that e2e delay increases with the number of flows because the interference among these flows will become more serious in given channels. On the other hand, more flows also increase traffic and throughput before the network becomes congested.

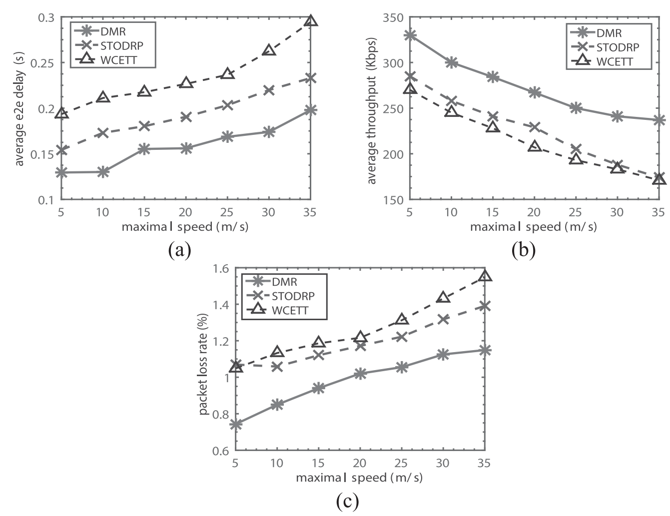


Fig. 10. Performance with the speed. (a) Average e2e delay. (b) Average throughput. (c) Packet loss rate.

5) Performance With Speed: In Fig. 10, we used different speeds to test three routing algorithms. As shown in Fig. 10, as the maximal speed increases, both e2e delay and packet loss rate increase, and throughput decreases. The reason is that our DMR considers node mobility but WCETT and STODRP do not take this consideration into account. So path duration in WCETT and STODRP is lower than that in our DMR. As a result, our DMR exhibits the best packet loss rate, throughput, and e2e delay in these proposals, shown in Fig. 10(a), (b), and (c) respectively.

6) Performance With Transmission Rate: Fig. 11 shows how performances change with packet transmission rates. Our DMR outperforms STODRP and WCETT in all scenarios. Specifically, our DMR exhibits lower e2e delay shown in Fig. 11(a) and packet loss rate shown in Fig. 11(c), and higher throughput shown in Fig. 11(b). Particularly, WCETT and STODRP both increase the delay apparently, and increase throughput slowly as transmission rate increases. The reason is that STODRP and WCETT cannot effectively avoid channel interference

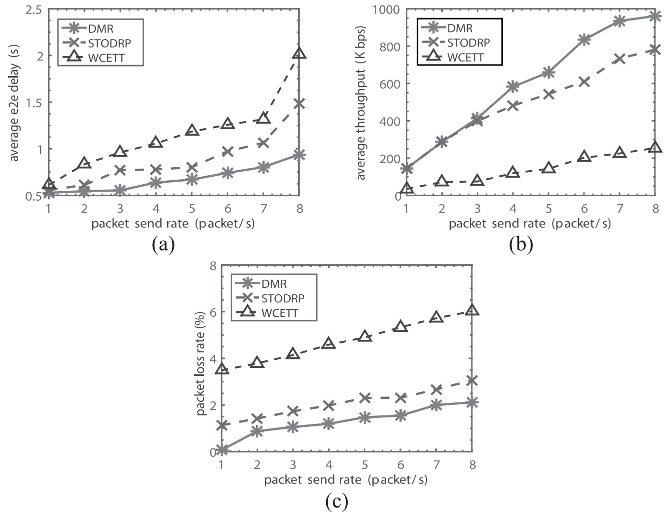


Fig. 11. Performance with the transmission rate. (a) Average e2e delay. (b) Average throughput. (c) Packet loss rate.

and do not calculate the media access delay efficiently. Instead, our DMR can predict the e2e delay precisely and assign conflict-free channels.

VII. Conclusion and Future Work

In this paper, we presented the DMR protocol that jointly selects routes and assigns channels based on the delay prediction in MCADNs. First, we propose an e2e delay model, which consists of the EMAT and expected transmission time, to predict an e2e delay in the collision probability. Second, we design a novel routing metric using the e2e path delay. Finally, we develop a heuristic routing algorithm that jointly explores routes with the minimal e2e delay and assigns channels in MCADNs for time-critical applications. NS2-based simulation results demonstrate that our DMR significantly outperforms related proposals in terms of average e2e delay, throughput, and packet loss rate.

As a part of our future work, we will improve our delay model through extending flow scheduling in intermediate nodes.

References

[1] T. Chiwewe, C. Mbuya, and G. Hancke, "Using cognitive radio for interference-resistant industrial wireless networks: An overview," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1466–1481, Dec. 2015.x

[2] A. S. Cacciapuoti, M. Caleffi, F. Marino, and L. Paura, "On the route priority for cognitive radio networks," *IEEE Trans. Commun.*, vol. 63, no. 9, pp. 3103–3117, Sep. 2015.

[3] J. Zhao, W. Gao, and Y. Wang, "Delay-constrained caching in cognitive radio networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2014, pp. 2094–2102.

[4] J. Wu, Y. Dai, and Y. Zhao, "Effective channel assignments in cognitive radio networks," *Comput. Commun.*, vol. 36, no. 4, pp. 411–420, 2013.

[5] I. Pefkianakis, S. Wong, and S. Lu, "SAMER: Spectrum aware mesh routing in cognitive radio networks," in *Proc. 3rd IEEE Symp. New Frontiers Dyn. Spectr. Access Netw.*, 2008, pp. 1–5.

[6] 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 Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1969–1979, May 2010.

[7] J. Zhang, F. Yao, Y. Liu, and L. Cao, "Robust route and channel selection in cognitive radio networks," in *Proc. IEEE 14th Int. Conf. Commun. Technol.*, 2012, pp. 202–208.

[8] G. Shah, V. C. Gungor, and O. B. Akan, "A cross-layer QOS-aware communication framework in cognitive radio sensor networks for smart grid applications," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1477–1485, Aug. 2013.

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

[10] 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 *Proc. IEEE Global Telecommun. Conf.*, Nov. 2008, pp. 1–5.

[11] G. Cheng, W. Liu, Y. Li, and W. Cheng, "Spectrum aware on-demand routing in cognitive radio networks," in *Proc. 2nd IEEE Symp. New Frontiers Dyn. Spectr. Access Netw.*, Apr. 2007, pp. 571–574.

[12] K. Chowdhury and M. Di Felice, "SEARCH: A routing protocol for mobile cognitive radio ad-hoc networks," in *Proc. IEEE Sarnoff Symp.*, Mar. 2009, pp. 1–6.

[13] M. Caleffi and A. S. Cacciapuoti, "Optimal database access for TV white space," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 83–93, Jan. 2016.

[14] X. J. Feng, Q. Zhang, and J. Zhang, "A hybrid pricing framework for TV white space database," *IEEE Trans. Wirel. Commun.*, vol. 13, no. 5, pp. 2626–2635, May 2014.

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

[16] K. M. Kang and J. C. Park, "A new scheme for compliance with TV white space regulations using Wi-Fi modules in a cognitive radio system," *IEEE Trans. Consumer Electron.*, vol. 60, no. 4, pp. 567–573, Nov. 2014.

[17] M. Fadda, M. Murroni, and V. Popescu, "A cognitive radio indoor HDTV multi-vision system in the TV white spaces," *IEEE Trans. Consumer Electron.*, vol. 58, no. 2, pp. 302–310, May 2012.

[18] A. El-Sherif and A. Mohamed, "Joint routing and resource allocation for delay minimization in cognitive radio based mesh networks," *IEEE Trans. Wirel. Commun.*, vol. 13, no. 1, pp. 186–197, Jan. 2014.

[19] H. Wu, F. Yang, K. Tan, J. Chen, Q. Zhang, and Z. Zhang, "Distributed channel assignment and routing in multiradio multichannel multihop wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 1972–1983, Nov. 2006.

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

[21] Q. S. Guan, F. R. Yu, S. M. Jiang, and G. Wei, "Prediction-based topology control and routing in cognitive radio mobile ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 9, pp. 4443–4452, Nov. 2010.

[22] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw.*, 2004, pp. 114–128.

[23] Z. Yu, H. Wang, B. Guo, T. Gu, and T. Meiet, "Supporting serendipitous social interaction using human mobility prediction," *IEEE Trans. Human–Mach. Syst.*, vol. 45, no. 6, pp. 811–818, Dec. 2015.

[24] Z. Yu, H. Xu, Z. Yang, and B. Guo, "Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints," *IEEE Trans. Human–Mach. Syst.*, vol. 46, no. 1, pp. 151–158, Feb. 2016.

[25] B. Guo, H. Chen, Z. Yu, X. Xie, S. Huang, and D. Zhang, "FlierMeet: a mobile crowdsensing system for cross-space public information reposting, tagging, and sharing," *IEEE Trans. Mobile Comput.*, vol. 14, no. 10, pp. 2020–2033, Oct. 2015.

Feilong Tang (SM'15) received the Ph.D. degree in computer science from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2005.

He is currently a Full Professor in the Department of Computer Science and Engineering, SJTU. In past years, he was the Japan Society for the Promotion of Science Research Fellow in Japan. His research interests include mobile cognitive network, big data analysis, clouding computing, and algorithm design and evaluation.

Dr. Tang received two best papers from international conferences and the Distinguished Pu-Jiang Scholars Award from Shanghai Municipality. He is an IET Fellow.

Can Tang is working toward the Master's degree in the College of Business and Economics, The Australian National University, Canberra, Australia.

Her current research interests include computational finance, financial big data analysis, and algorithm design and evaluation.

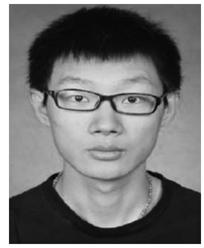
Yanqin Yang received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2011.

She is currently an Associate Professor in the Department of Computer Science and Technology, East China Normal University, Shanghai, China. Her research interests include computer network and embedded system.

Laurence T. Yang (SM'15) received the Ph.D. degree in computer science from the University of Victoria, Victoria, BC, Canada.

He is a Professor in the Department of Computer Science, St. Francis Xavier University, Antigonish, NS, Canada. His research interests include parallel and distributed computing, embedded and ubiquitous/pervasive computing, and big data.

Dr. Yang has published more than 200 papers in various refereed journals.



Tong Zhou received the B.E. degree in software engineering from the Northwestern Polytechnical University, Xi'an, China, in 2014. He is working toward the M.S. degree in software engineering at Shanghai Jiao Tong University, Shanghai, China.

His research interests include wireless network and mobile computing.

Jie Li (SM'04) received the Dr. Eng. degree from the University of Electro-Communications, Tokyo, Japan.

He has been with the University of Tsukuba, Tsukuba, Japan, where he is a Professor. His research interests include mobile distributed multimedia computing and networking, OS, network security, modeling, and performance evaluation of information systems.

Minyi Guo (SM'07) received the Ph.D. degree in computer science from the University of Tsukuba, Tsukuba, Japan.

He is currently a Zhiyuan Chair Professor in Shanghai Jiao Tong University, Shanghai, China. His research interests include pervasive computing, parallel and distributed processing, and parallelizing compilers.

In 2007, he received the Recruitment Program of Global Experts and Distinguished Young Scholars Award from the National Natural Science Foundation of China.