J Comb Optim (2013) 26:161–177
DOI 10.1007/s10878-011-9446-7

Practical unicast and convergecast scheduling schemes for cognitive radio networks

Shouling Ji · Arif Selcuk Uluagac · Raheem Beyah · Zhipeng Cai

Published online: 10 January 2012

Abstract Cognitive Radio Networks (CRNs) have paved a road for Secondary Users (SUs) to opportunistically exploit unused spectrum without harming the communications among Primary Users (PUs). In this paper, practical unicast and convergecast schemes, which are overlooked by most of the existing works for CRNs, are proposed. We first construct a cell-based virtual backbone for CRNs. Then prove that SUs have positive probabilities to access the spectrum and the expected one hop delay is bounded by a constant, if the density of PUs is finite. According to this fact, we proposed a three-step unicast scheme and a two-phase convergecast scheme. We demonstrate that the induced delay from our proposed *Unicast Scheduling* (US) algorithm scales linearly with the transmission distance between the source and the destination. Furthermore, the expected delay of the proposed *Convergecast Scheduling* (CS) algorithm is proven to be upper bounded by $O(\log n + \sqrt{n/\log n})$. To the best of our knowledge, this is the first study of convergecast in CRNs. Finally, the performance of the proposed algorithms is validated through simulations.

Keywords Cognitive radio networks · Unicast · Convergecast

S. Ji · Z. Cai (✉)
Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA
e-mail: zcai@cs.gsu.edu

S. Ji
e-mail: sji@cs.gsu.edu

A.S. Uluagac · R. Beyah
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

A.S. Uluagac
e-mail: selcuk@ece.gatech.edu

R. Beyah
e-mail: rbeyah@ece.gatech.edu

1 Introduction

The inefficient utilization of licensed spectrum and crowded unlicensed spectrum necessitate a new communication paradigm, named Cognitive Radio Network (CRN). In CRNs, unlicensed users (Secondary Users) can identify and utilize locally unused portion of the spectrum without causing unacceptable interference to the licensed users (Primary Users) (Akyildiz et al. 2006). Numerous efforts have been concentrating on single-hop scenarios, including spectrum sensing (Zhang and Tsang 2011; Kim and Shin 2008), spectrum decision, and spectrum sharing (Huang et al. 2009; Wang et al. 2008; Shi and Hou 2008; Shu and Krunz 2009; Kasbekar and Sarkar 2010) techniques. Most recently, researchers have realized that multi-hop CRNs have great potential to improve the spectrum utilization for secondary users. Among those multi-hop CRNs studies, capacity/throughput/delay scaling laws (Jeon et al. 2008; Wang et al. 2010; Yin et al. 2010; Li and Dai 2011a, 2011b; Huang and Wang 2011; Kompella et al. 2011; Sun and Wang 2011; Han and Yang 2011), network connectivity (Ren et al. 2010; Wang et al. 2011), routing protocols (Huang et al. 2011; Shu and Krunz 2010; Pan et al. 2011), and multicast communication (Jin et al. 2010; Hu et al. 2009) have attracted a lot of attention.

Unicast, multicast, and convergecast are important communication modes under the multi-hop scenarios. Unicast is a data transmission between *data source* and *data destination* via multi-hops, multicast is to transmit data from a *data source* to a group of *data destinations*, while convergecast is to gather data packets produced by all of the SUs at a particular time slot to the *sink* (*base station*). There are quite a few works related to unicast and multicast in CRNs (Ren et al. 2010; Wang et al. 2011; Jin et al. 2010; Hu et al. 2009). Most of them focused on the capacity, throughput, and delay scaling issues under different assumptions. To name some, Jeon et al. (2008) showed primary and secondary users can simultaneously achieve the same throughput scaling law if the secondary network is denser than the primary network and the primary network throughput is subject to a fractional loss. Wang et al. (2010) designed an optimal-throughput strategy for secondary networks. They showed there exists a threshold of the density of SUs such that the secondary network can achieve the multicast capacity of the same order as its stand-alone case. Yin et al. (2010) demonstrated that both primary and secondary networks can achieve the same capacity and delay-throughput scaling laws under certain assumptions. Similarly, Huang and Wang (2011) proposed a hybrid protocol model. The secondary network under this model can achieve the same throughput and delay scaling as a stand alone network in the round-robin TDMA manner. Kompella et al. (2011) studied the stable throughput tradeoffs in CRNs with cooperative relaying. Sun and Wang (2011) demonstrated that the dissemination latency depends on the stationary spatial distribution and the mobility capacity of secondary users. Another group of research related to unicast concentrated on analyzing the CRN connectivity. Dousse et al. (2003) studied the connectivity of ad hoc networks via the percolation theory. Following the same strategy, the authors in Ren et al. (2010), Wang et al. (2011) exploited the theories and techniques from continuum percolation and ergodicity. They showed that the scaling law of the minimum multihop delay is related to the source-destination distance.

Most existing works that address the unicast, broadcast, or/and multicast issues in CRNs only focus on the theoretical aspects, especially the scaling laws, under some assumptions and network conditions. The lack of practical unicast schemes becomes a bottleneck for real applications to make maximum use of this emerging communication paradigm. Furthermore, there are no solutions dedicated for convergecast in CRNs. Therefore, designing practical unicast and convergecast schemes and theoretically analyzing their corresponding delay bounds are desirable. In this paper, we partially fill this gap by proposing practical unicast and convergecast schemes for CRNs. Meanwhile, we theoretically analyze the delay bounds of the proposed schemes.

In a CRN, a transmission between two SUs not only suffers from the interference caused by other concurrent communications in the secondary network, but also is affected by the activities of the primary users. Thus, the management and scheduling of the secondary network are more challenging. Constructing *virtual backbones*, which has brought significant benefits for conventional wireless networks, is a potential solution to manage CRNs (Li et al. 2006; He et al. 2011). By exploiting a virtual backbone, the routing and topology control issues become much easier, as well as less communication and storage overhead are incurred. In this paper, a cell based virtual backbone for the secondary network is constructed. We prove that (i) if $N/A \leq \infty$, where N is the number of PUs in the primary network and A is the size of the deployed area of the primary network, then the SUs have positive probabilities to access the spectrum; and (ii) the expected one hop delay in the constructed virtual backbone is bounded by a constant. Our unicast and convergecast scheduling are designed based on such a virtual backbone. In our *Unicast Scheduling* (US) algorithm, the induced delay scales linearly with the distance between the data source and the data destination, which is consistent with the theoretical scaling laws in Ren et al. (2010), Wang et al. (2011). In the *Convergecast Scheduling* (CS) algorithm, the cells are re-grouped to form *concurrent cell sets*, that is, multiple cells can be scheduled to obtain local aggregation values concurrently. These local aggregation values will be transmitted to the sink level by level along the convergecast tree. Theoretical analysis shows that the expected delay is upper bounded by $O(\log n + \sqrt{n/\log n})$. We evaluate performance of our proposed work by simulations and the results demonstrate that the proposed algorithms can significantly reduce the incurred unicast and convergecast delay.

The rest of the paper is organized as follows. In Sect. 2, we give the network model. In Sect. 3, the cell-based virtual backbone is constructed. In Sects. 4 and 5, the virtual backbone based unicast and convergecast schemes are designed and analyzed, respectively. In Sect. 6, the performance and effectiveness of the proposed algorithms are verified. Finally, we conclude this paper in Sect. 7.

2 Preliminaries

In this paper, we consider a dense secondary network which coexists with a primary network. Both of the networks are deployed in a square area $\mathcal{A}$ and they share the same time, space, and spectrum.

Primary network For a primary network, $N < \infty$ Primary Users (PUs) (*licensed users*), denoted by $S_1, S_2, \ldots, S_N$, are *independent and identically distributed* (*i.i.d.*)

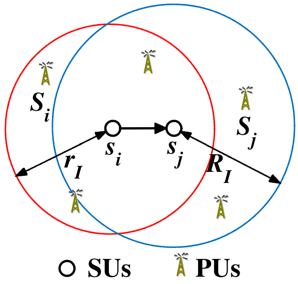


Fig. 1 Illustration of the communication between two secondary users, s_i and s_j

in $\mathcal{A}$. The transmission radius of S_i $(1 \le i \le N)$ is set to R and the interference radius of S_i $(1 \le i \le N)$ is $R_I = \rho_p \cdot R$, where $\rho_p \ge 1$. τ is defined as a unit of network time. Furthermore, a data packet can be successfully transmitted in a unit network time as long as there is no interference.

A generalized probabilistic model is adopted to describe the activities of the primary network. During a particular time slot, S_i $(1 \le i \le N)$ has probability p_t $(0 \le p_t \le 1)$ to transmit data and p_r $(0 \le p_r \le 1)$ to receive data. Note that p_t and p_r can be determined accordingly (Ross 2007) under different probabilistic distributions, such as the Poisson distribution and the Uniform distribution.

Secondary network In a secondary network, $n > N$ Secondary Users (SUs) (unlicensed users), denoted by $s_1, s_2, \ldots, s_n$, are *i.i.d.* in $\mathcal{A}$. Each SU is equipped with one radio and works with a fixed power. The transmission radius of s_i $(1 \le i \le n)$ is r and the interference radius is $r_I = \rho_s \cdot r$, where $\rho_s \ge 1$. Additionally, we assume the transmission radius of PUs is $R = c \cdot r$, where $c > 0$ is a constant. There is a *logical link* (edge) between s_i and s_j *iff* $\|s_i - s_j\| \le r$, where $\|\cdot\|$ is the Euclidean distance between them. Clearly, the topological structure of the secondary network can be modeled by a graph $G_s = (V_s, E_s)$, where $V_s = \{s_1, s_2, \ldots, s_n\}$ is the node set, and E_s is the *logical link set*.

Since PUs have priority to utilize the spectrum, a logical link between s_i and s_j does not imply that s_i and s_j can conduct communication. From Fig. 1, even there is a logical link between s_i and s_j, s_i cannot successfully transmit data to s_j if (i) a PU located at the circle centered at s_i with radius r_I is receiving some data, *e.g.* S_i in Fig. 1, or (ii) a PU which is located at the circle centered at s_j with radius R_I is transmitting some data, *e.g.* S_j. In order to accurately define the possible conducted communications, we define two events as follows, where $x \in \{s_1, s_2, \ldots, s_n\}$, $y \in \mathbb{R}^+$, $tx =$ *transmitting some data*, and $rx =$ *receiving some data*.

Definition 1 ($\mathbf{E}(x, y, tx)$ and $\mathbf{E}(x, y, rx)$) $\mathbf{E}(x, y, tx)$ (respectively, $\mathbf{E}(x, y, rx)$) is the event that within the circle centered at x of radius y, there is at least one PU transmitting (respectively, receiving) some data.

Based on Definition 1, we define a directional *physical link* from s_i to s_j as follows.

Definition 2 (Physical link) The link from s_i to s_j, denoted by (s_i, s_j), is a physical link if (i) there is a logical link in E_s between s_i and s_j; (ii) $\overline{\mathbf{E}(s_i, r_I, rx)} = \mathbf{true}$,

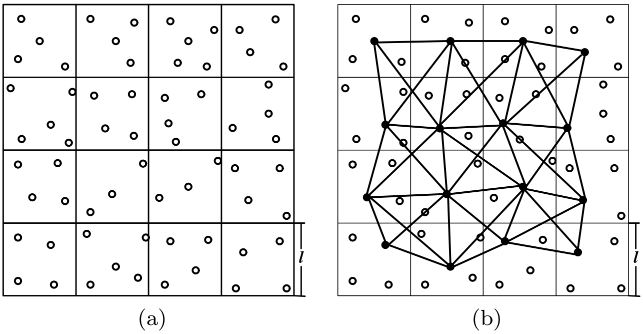


Fig. 2 A cell based virtual backbone for the secondary network. In (**b**), the *black* SUs are selected *backbone nodes* and the other SUs are *non backbone nodes*. The links from the *non backbone nodes* to *backbone nodes* are omitted for clarity

where $\overline{\mathbf{E}(s_i, r_I, rx)}$ is the *complement event* of $\mathbf{E}(s_i, r_I, rx)$; and (iii) $\overline{\mathbf{E}(s_j, R_I, tx)} =$ **true**.

In Definition 2, the second condition implies the transmission of s_i does not interfere any PU which is receiving data. The third condition implies the data reception at s_j is not interfered by any PU which is transmitting data. In the current definition, s_i has a spectrum opportunity to transmit data to s_j does not imply s_j also has such a spectrum opportunity. However, some network protocols require the receiver to send back a short *acknowledgement* packet after successfully receiving a data packet. Then we define a bidirectional *guaranteed physical link* for those protocols.

Definition 3 (Guaranteed physical link) The link between s_i and s_j is a guaranteed physical link if (i) there is a logical link in E_s between s_i and s_j; (ii) link (s_i, s_j) and (s_j, s_i) are physical links.

3 Cell-based virtual backbone

3.1 Network partition

We construct a *cell-based virtual backbone* for the secondary network, which is *i.i.d.* in a square $\mathcal{A}$ with size A. The network is partitioned into square *cells* with side length $l = \sqrt{\frac{3A\log n}{n}}$. Without loss of generality, we assume $h = \sqrt{A}/l$ is an integer. Additionally, we assume the communication radius of a SU $r = 2\sqrt{2}l$, which implies the SUs in a cell can transmit data to the SUs in the neighboring cells via one hop. For example, we partition the secondary network shown in Fig. 2(a) into 16 cells. Each cell is assigned coordinates (i, j) $(1 \le i, j \le h)$ for easy presentation, where i and j indicate this cell is located at the i-th column and j-th row respectively ((1, 1) is assigned to the left-bottom-most cell). Furthermore, we use $\mathfrak{S}(\mathcal{C}_{i,j})$ to denote the set of SUs locating in cell $\mathcal{C}_{i,j}$.

Then, the following lemma can be obtained.

Lemma 1 *Let Z be a variable representing the number of SUs within any cell. Then, the following statements are true*: (i) *Z satisfies the binomial distribution with parameters $(n, \frac{l^2}{A})$ and the expectation of Z $\mathbb{E}[Z] = 3\log n$*; (ii) *it is almost sure that* $Z \le 8\log n$, i.e. *it is almost sure no cell contains more than $8\log n$ SUs*; *and* (iii) *it is almost sure that $Z \ge \log n/5$*.

Proof (i) Since all the SUs are *i.i.d.* in $\mathcal{A}$, the number of SUs in a cell satisfies the *binomial distribution* with parameters $(n, \frac{l^2}{A})$ and $\mathbb{E}[Z] = n \cdot \frac{l^2}{A} = 3\log n$.

(ii) Applying the Chernoff bound and for any $\gamma > 0$, we have

$$\Pr(Z \ge 8\log n) \le \min_{\gamma>0} \frac{\mathbb{E}[e^{\gamma Z}]}{e^{8\gamma \log n}} = \min_{\gamma>0} \frac{(l^2 e^{\gamma}/A + 1 - l^2/A)^n}{e^{8\gamma \log n}} \tag{1}$$

$$= \min_{\gamma>0} e^{(3e^{\gamma}-3-8\gamma)\log n}. \tag{2}$$

Let $\gamma = 1$, we have

$$\Pr(Z \ge 8\log n) \le e^{(3e-11)\log n} \le e^{-2\log n} \le e^{-2\ln n} = \frac{1}{n^2}. \tag{3}$$

$\sum_{n\ge1} \frac{1}{n^2} = \frac{\pi^2}{6}$ is the Riemann Zeta function with parameter 2. According to the Borel–Cantelli Lemma, $\Pr(Z \ge 8\log n) \sim 0$, *i.e.* it is almost sure that the number of SUs within any cell is no more than $8\log n$.

(iii) Similarly, applying the Chernoff bound and for any $\gamma < 0$, we have $\Pr(Z \le \frac{\log n}{5}) \le \min_{\gamma<0} \frac{\mathbb{E}[e^{\gamma Z}]}{e^{\gamma \log n/5}} \le \frac{1}{n^2}$ $(\gamma = -2)$. Therefore, statement (iii) is true. □

Based on Lemma 1, it is reasonable to take $8\log n$ and $\log n/5$ as the upper and lower bounds respectively of the number of SUs in a cell.

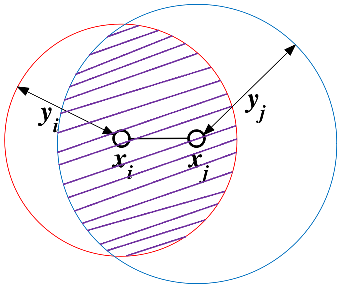
3.2 Virtual backbone construction

After partitioning the secondary network into cells, we randomly choose a SU in each cell as the *backbone node*. For convenience, let $\mathfrak{B}(\mathcal{C}_{i,j})$ be the selected backbone node in cell $\mathcal{C}_{i,j}$. Subsequently, a virtual backbone for the secondary network can be constructed by (i) connecting the backbone nodes within transmission range; and (ii) connecting all the *non backbone nodes* and the backbone node within the same cell. For the secondary network in Fig. 2(a), a virtual backbone is constructed in Fig. 2(b).

Before studying the one-hop delay for the virtual backbone, Lemma 2 is proven first which calculates the probability that the number of PUs in an area with size Λ is above a bound.

Lemma 2 *Let Z be the random variable representing the number of PUs in an area with size Λ. Then, for any $\gamma > 0$, $\Pr(Z \ge a) \le \min_{\gamma>0} \exp[\frac{N\Lambda}{A}(e^{\gamma} - 1) - \gamma a]$.*

Fig. 3 The intersection part of two circles



Proof Similar as the proof of Lemma 1, Z is a random variable satisfying the *binomial distribution* with parameters $(N, \frac{\Lambda}{A})$. Therefore, applying the Chernoff bound, we obtain $\Pr(Z \geq a) \leq \min_{\gamma>0} \frac{\mathbb{E}[e^{\gamma Z}]}{e^{\gamma a}} \leq \min_{\gamma>0} \exp[\frac{N\Lambda}{A}(e^{\gamma}-1)-\gamma a]$. □

Lemma 3 *Let* $\phi = \frac{24\pi\rho_s^2 N \log n}{n}$, $\varphi = \frac{24\pi c^2 \rho_p^2 N \log n}{n}$, $\alpha = \frac{\phi - 2\log n}{1-\ln\phi^{-1}}$, $\beta = \frac{\varphi - 2\log n}{1-\ln\varphi^{-1}}$, $N < \infty$ *and* $n \to \infty$. *Then,* (i) *it is almost sure that the number of PUs within a circle with radius* r_I *is no more than* α, *where* α *is a constant value*; (ii) *it is almost sure that the number of PUs within a circle with radius* R_I *is no more than* β, *where* β *is a constant value.*

Proof (i) The area of a circle with radius r_I is $\pi r_I^2 = \pi\rho_s^2 r^2 = \frac{24\pi\rho_s^2 A \log n}{n}$. Let Z be the number of PUs within this circle. Then, from Lemma 2, we have

$$\Pr(Z \geq \alpha) \leq \min_{\gamma>0} \exp\left[\frac{N}{A} \cdot \frac{24\pi\rho_s^2 A \log n}{n}(e^{\gamma}-1) - \gamma\alpha\right] \tag{4}$$

$$= \min_{\gamma>0} \exp[\phi(e^{\gamma}-1) - \gamma\alpha]. \tag{5}$$

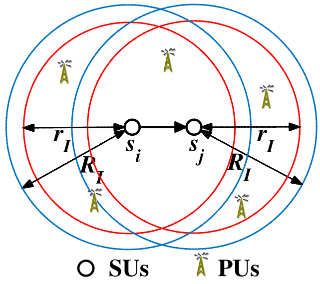
Let $\gamma = \ln\frac{\alpha}{\phi}$. Then, $\Pr(Z \geq \alpha) \leq \exp(\alpha - \phi - \alpha\ln\frac{\alpha}{\phi})$. When $n \to \infty$ and $N < \infty$, $\phi = O(\frac{\log n}{n}) \to 0$, $\phi^{-1} = O(\frac{n}{\log n}) \to \infty$, $\ln\phi^{-1} = O(\ln\frac{n}{\log n}) \to \infty$, and $\alpha = O(\frac{\log n}{\ln(n/\log n)}) \to$ *constant*. It follows that $\Pr(Z \geq \alpha) \sim 0$, that is, it is almost sure that the number of PUs within a circle with radius r_I is no more than a constant α.

(ii) Similar as the proof in (i), we can prove that $\Pr(Z \geq \beta) \sim 0$. □

Let $x_i, x_j \in V_s$ and $y_i, y_j \in \mathbb{R}^+$ ($\mathbb{R}^+$ is the set of positive real numbers). Define $\mathcal{S}(x_i, y_i, x_j, y_j)$ as the *intersection part* of the circle centered at x_i with radius y_i and the circle centered at x_j with radius y_j. For instance, the shaded area is the intersection part of the two shown circles (Fig. 3). Then, for $\forall s_i, s_j \in V_s$, suppose there is a logical link between s_i and s_j in the constructed virtual backbone. We can obtain the one-hop delay for a data transmission between s_i and s_j as shown in Theorem 1.

Theorem 1 (i) *During a time slot, the probability* p_p *that there is a physical link from* s_i *to* s_j *satisfies* $p_p \geq (1-p_r)^{\alpha} \cdot (1-p_t)^{\beta}$. *It follows that the expected delay*

Fig. 4 The spectrum opportunity for a guaranteed physical link



of a data transmission from s_i to s_j is less than τ/p_p; (ii) *during a time slot, the probability p_g that a guaranteed physical link formed between s_i and s_j satisfies $p_g \geq (1-p_r)^{2\alpha-\alpha_c} \cdot (1-p_t)^{2\beta-\beta_c}$, where α_c is the number of PUs in $\mathcal{S}(s_i, r_I, s_j, r_I)$ and β_c is the number of PUs in $\mathcal{S}(s_i, R_I, s_j, R_I)$. The expected delay of a guaranteed data transmission between s_i and s_j is less than τ/p_g.*

Proof (i) Based on Lemma 3, $p_p = \Pr[$*a logical link between s_i and s_j changes to a physical link*$] \geq (1-p_r)^{\alpha} \cdot (1-p_t)^{\beta}$. That is, the expected delay of a data transmission from s_i to s_j is upper bounded by τ/p_p.

(ii) As shown in Fig. 4, the receiving activities of PUs locating at $\mathcal{S}(s_i, r_I, s_j, r_I)$ may be interfered by both s_i and s_j. On the other hand, the transmitting activities of PUs locating at $\mathcal{S}(s_i, R_I, s_j, R_I)$ may interfere both s_i and s_j. Let α_c and β_c be the numbers of PUs within $\mathcal{S}(s_i, r_I, s_j, r_I)$ and $\mathcal{S}(s_i, R_I, s_j, R_I)$, respectively. Then, $p_g = \Pr[$*a logical link between s_i and s_j changes to a guaranteed physical link*$] \geq (1-p_r)^{2\alpha-\alpha_c} \cdot (1-p_t)^{2\beta-\beta_c}$. It follows that the expected delay of a guaranteed data transmission between s_i and s_j is upper bounded by τ/p_g. □

From Theorem 1, we know that (i) if $N/A \leq \infty$, the SUs have positive probabilities to access the spectrum; and (ii) if the SUs have positive probabilities to access the spectrum, the expected one hop delay in the constructed virtual backbone is bounded by a constant.

4 Virtual backbone based unicast

In this section, we design a *Unicast Scheme* (US) based on the constructed virtual backbone and analyze the delay performance of US. First, we formally define the *unicast problem* as follows.

Definition 4 (Unicast problem) The unicast problem deals with the problem of transmitting one data packet from a data source $s_i \in V_s$ to a data destination $s_j \in V_s$ over multiple hops.

For a unicast task, assume $s_u \in \mathcal{C}_{i,j}$ is the data source and $s_v \in \mathcal{C}_{i',j'}$ is the data destination. Then, a Unicast Scheme (US) based on the constructed virtual backbone is

Algorithm 1 The unicast scheme

Require: the virtual backbone of the secondary network, the data source s_u, the destination s_v

1: s_u transmits the data packet to $\mathfrak{B}(\mathcal{C}_{i,j})$

2: construct a shortest data transmission path from $\mathfrak{B}(\mathcal{C}_{i,j})$ to $\mathfrak{B}(\mathcal{C}_{i',j'})$ through the virtual backbone, denoted by $\mathfrak{B}(\mathcal{C}_{i,j}) \leftrightarrow \mathfrak{B}_2(\cdot) \leftrightarrow \cdots \leftrightarrow \mathfrak{B}_{H-2}(\cdot) \leftrightarrow \mathfrak{B}(\mathcal{C}_{i',j'})$

3: $\mathfrak{B}(\mathcal{C}_{i,j})$ forwards the data packet to $\mathfrak{B}(\mathcal{C}_{i',j'})$ along the constructed shortest path

4: $\mathfrak{B}(\mathcal{C}_{i',j'})$ forwards the data packet to s_v

shown in Algorithm 1. In Algorithm 1, the constructed path from s_u to s_v is assumed to be H hops.

Since we have analyzed the one hop delay performance for the constructed virtual backbone, the following Theorem can be obtained.

Theorem 2 *Let H be the number of hops from s_u to s_v in US and $D(s_u, s_v)$ be the physical distance between s_u and s_v. Then*, (i) *if the unicast transmissions are over physical links*, *the expected delay of US is less than $\frac{H\tau}{p_p}$ and $\exists 0 < \xi < \infty$ such that it is almost sure*

$$\lim_{D(s_u,s_v)\to\infty} \frac{H\tau}{p_p D(s_u, s_v)} = \xi; \tag{6}$$

(ii) *if the unicast transmissions are over guaranteed physical links*, *the expected delay of US is less than $\frac{H\tau}{p_g}$ and $\exists 0 < \xi < \infty$ such that it is almost sure*

$$\lim_{D(s_u,s_v)\to\infty} \frac{H\tau}{p_g D(s_u, s_v)} = \xi. \tag{7}$$

Proof (i) From Theorem 1, it is straightforward that the expected delay of US is less than $\frac{H\tau}{p_p}$ if the communications are conducted over physical links. Furthermore, according to the construction process of the cell-based virtual backbone, from any backbone node $\mathfrak{B}(\mathcal{C}_{i,j})$ to any other backbone node $\mathfrak{B}(\mathcal{C}_{i',j'})$, the number of hops of a shortest path is upper bounded by $\lceil \frac{\sqrt{A}}{l} \rceil = h$. It follows that the number of hops from s_u to s_v in US in upper bounded by $h + 2$. Additionally, $D(s_u, s_v) \to \infty \Leftrightarrow A \to \infty \Leftrightarrow H \le h + 2 \to \infty$. Therefore, $\exists 0 < \xi < \infty$ such that it is almost sure $\lim_{D(s_u,s_v)\to\infty} \frac{H\tau}{p_p D(s_u,s_v)} = \xi$.

(ii) By a similar method as in (i), the conclusions in (ii) can be proven. □

From Theorem 2, when SUs have a positive probability to access the spectrum, then the induced delay of US scales linearly with the distance between the data source and the data destination, which is consistent with the theoretical scaling laws in Ren et al. (2010), Wang et al. (2011).

5 Virtual backbone based convergecast

5.1 Problem definition

At a particular time instant, we assume each SU produces a data packet and the union of all the data packets is called a *snapshot*. In this section, we study the virtual backbone based convergecast scheduling, where multiple data packets can be aggregated into one data packet by some aggregation function, *e.g.* MIN, MAX, SUM, and the final aggregation value of a snapshot will be transmitted to the *sink* (*base station*).

Let X and Y be two subsets of V_s and $X \cap Y = \emptyset$. The data of the SUs in X is said to be *aggregated* to the SUs in Y during a time slot if (i) there are spectrum opportunities for the SUs in X, and (ii) all the SUs in X can transmit their data packets to the SUs in Y simultaneously and interference-freely during a time slot. Then, the *convergecast problem* can be formally defined as follows.

Definition 5 (Convergecast problem) We seek a convergecast schedule which is a sequent of transmitter sets $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_m$ such that (i) $\forall i \neq j$, $\mathfrak{A}_i \cap \mathfrak{A}_j = \emptyset$; (ii) $\bigcup_{i=1}^{m} \mathfrak{A}_i = V_s \setminus \{\mathfrak{s}\}$, where $\mathfrak{s}$ is a SU which is serving as the sink; and (iii) data can be aggregated from $\mathfrak{A}_j$ to $V_s \setminus \bigcup_{i=1}^{j} \mathfrak{A}_i$ at time slot j for $j = 1, 2, \ldots, m$ and the final aggregation value of a snapshot can be obtained at $\mathfrak{s}$ during time slot m.

5.2 Concurrent cell set

When seek the data aggregation schedule, we want as many SUs as possible to initiate data transmissions simultaneously as long as they have spectrum opportunities. Let $d \in \mathbb{R}^+$ be a positive constant and $\mathbb{C}$ be a set of SUs such that $\forall s_i, s_j \in \mathbb{C}$, if $s_i \neq s_j$, $\|s_i - s_j\| = b \cdot d$, where $b \in \mathbb{Z}^+$ is a positive integer. When all the SUs in $\mathbb{C}$ have spectrum opportunities simultaneously, the following Lemma 4 shows the condition on d that all the SUs in $\mathbb{C}$ can conduct data transmissions concurrently.

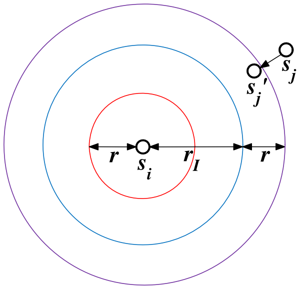
Lemma 4 *If $d > r + r_I$, then all the nodes in $\mathbb{C}$ can conduct data transmissions as long as the spectrum is available.*

Proof Suppose the spectrum is available for all the SUs in $\mathbb{C}$ and they conduct data transmissions simultaneously. Then, $\forall s_i, s_j \in \mathbb{C}$, assuming the receiver of s_j is s'_j. According to the network model defined in Sect. 2, if $d > r + r_I$, then $\|s_i - s'_j\| > r_I$ as shown in Fig. 5. Therefore, s_i does not interfere s'_j. By the same method, we can show all the transmissions initiated by the SUs in $\mathbb{C}$ are interference free, *i.e.* the lemma is true. □

Similar as Lemma 4, the following lemma, which states the condition on d that all the SUs in $\mathbb{C}$ can conduct guaranteed data transmissions concurrently, can be derived.

Lemma 5 *If $d > 2r + r_I$, then all the nodes in $\mathbb{C}$ can conduct data transmissions as long as the spectrum is available.*

Fig. 5 The distance between s_i and s'_j



Following Lemmas 4 and 5, let $d_1 = \lceil \frac{r+r_I+\epsilon}{l} \rceil$ and $d_2 = \lceil \frac{2r+r_I+\epsilon}{l} \rceil$, where ϵ is an arbitrary small positive value. Furthermore, define $\mathbb{S}_{i,j} = \{\mathcal{C}_{i',j'} | i' \in \{i + k \cdot (d_1 + 1) : k \in \mathbb{Z}\}, j' \in \{j + k \cdot (d_1 + 1) : k \in \mathbb{Z}\}, 1 \le i', j' \le h\}$ and $\mathbb{S}'_{i,j} = \{\mathcal{C}_{i',j'} | i' \in \{i + k \cdot (d_2 + 1) : k \in \mathbb{Z}\}, j' \in \{j + k \cdot (d_2 + 1) : k \in \mathbb{Z}\}, 1 \le i', j' \le h\}$, where $\mathbb{Z}$ is the integer set. Then, the following Corollary 1, which states the cells that can conduct data transmissions concurrently, can be obtained.

Corollary 1 *All the cells in $\mathbb{S}_{i,j}$ can conduct data transmissions concurrently without interference as long as they have spectrum opportunities. All the cells in $\mathbb{S}'_{i,j}$ can conduct guaranteed data transmissions concurrently without interference as long as they have spectrum opportunities.*[1]

For convenience, we call $\mathbb{S}_{i,j}$ (respectively, $\mathbb{S}'_{i,j}$) a *concurrent cell set* (respectively, *guaranteed concurrent cell set*). Let $\kappa = \min\{d_1 + 1, h\}$ (respectively, $\kappa' = \min\{d_2 + 1, h\}$). Then, according to Corollary 1, the h^2 cells of the secondary network can be partitioned into κ^2 *concurrent cell sets* $\mathbb{S}_{i,j}$ $(1 \le i, j \le \kappa)$ (respectively, κ'^2 *guaranteed concurrent cell sets* $\mathbb{S}'_{i,j}$ $(1 \le i, j \le \kappa')$) with the SUs in each concurrent cell set being able to conduct data transmissions (respectively, guaranteed data transmissions) simultaneously without interference. For example, if $d_1 = 2$ and $h = 8$, the network shown in Fig. 6 can be partitioned into 9 concurrent cell sets.

5.3 Convergecast scheduling

According to the defined (guaranteed) concurrent cell sets, we propose a virtual backbone based *Convergecast Scheduling* (CS) algorithm in this subsection. Without loss of generality and loosing only a constant factor, we assume $\mathfrak{s} = \mathfrak{B}(\mathcal{C}_{h,h})$, *i.e.* $\mathfrak{B}(\mathcal{C}_{h,h})$ is the sink. Furthermore, for simplicity, we define $\mathcal{L}_i = \{\mathcal{C}_{i,j} : i \le j \le h\} \cup \{\mathcal{C}_{j,i} : i \le j \le h\}$ for $(1 \le i \le h)$. Then, we construct a directional convergecast tree $\mathcal{T}$ with the following steps: (i) in each cell $\mathcal{C}_{i,j}$, all the SUs in $\mathfrak{S}(\mathcal{C}_{i,j}) \setminus \{\mathfrak{B}(\mathcal{C}_{i,j})\}$ transmit their data packets to $\mathfrak{B}(\mathcal{C}_{i,j})$, *i.e.* connect all the SUs in $\mathfrak{S}(\mathcal{C}_{i,j}) \setminus \{\mathfrak{B}(\mathcal{C}_{i,j})\}$ with $\mathfrak{B}(\mathcal{C}_{i,j})$ as their parent node; (ii) connect $\mathfrak{B}(\mathcal{C}_{i,j})$ $(1 \le i, j \le h-1)$ with $\mathfrak{B}(\mathcal{C}_{i+1,j+1})$ as its

[1] In this paper, a cell conducts a data transmission (respectively, guaranteed data transmission) means a SU in this cell conducts a data transmission (respectively, guaranteed data transmission) to another SU in this cell or in neighboring cells.

Fig. 6 Concurrent cell sets of a secondary network ($h = 8, d_1 = 2$). The number inside each cell indicates the concurrent cell set it belongs to. The cells with the same number come from the same concurrent cell set

4	5	6	4	5	6	4	5
1	2	3	1	2	3	1	2
7	8	9	7	8	9	7	8
4	5	6	4	5	6	4	5
1	2	3	1	2	3	1	2
7	8	9	7	8	9	7	8
4	5	6	4	5	6	4	5
1	2	3	1	2	3	1	2

parent node; (iii) connect $\mathfrak{B}(\mathcal{C}_{h,j})$ $(1 \le j \le h-1)$ with $\mathfrak{B}(\mathcal{C}_{h,j+1})$ as its parent node; and (iv) connect $\mathfrak{B}(\mathcal{C}_{i,h})$ $(1 \le i \le h-1)$ with $\mathfrak{B}(\mathcal{C}_{i+1,h})$ as its parent node.

Based on the convergecast tree $\mathcal{T}$, the CS is shown in Algorithm 2. Algorithm 2 consists of two phases. The first phase is the local data aggregation scheduling phase, where all the non backbone nodes in each cell transmit their data packets to their connected backbone nodes. The second phase is the virtual backbone based convergecast scheduling phase, where all the backbone nodes transmit the local aggregation values to their parents level by level.

Algorithm 2 The convergecast scheme

Require: the secondary network, the virtual backbone, the sink

{**Phase I: Local Data Aggregation Scheduling**}

for $i = 1; i \le \kappa/\kappa'; i++$ **do**

  for $j = 1; j \le \kappa/\kappa'; j++$ **do**

    schedule all the cells in $\mathbb{S}_{i,j}$ $(\mathbb{S}'_{i,j})$ simultaneously: $\forall \mathcal{C}_{u,v} \in \mathbb{S}_{i,j}$ $(\mathbb{S}'_{i,j})$, all the SUs in $\mathfrak{S}(\mathcal{C}_{u,v}) \setminus \{\mathfrak{B}(\mathcal{C}_{u,v})\}$ transmit their data packets to $\mathfrak{B}(\mathcal{C}_{u,v})$ sequentially, until $\mathfrak{B}(\mathcal{C}_{u,v})$ has obtained the final local aggregation value of all the data packets produced in cell $\mathcal{C}_{u,v}$

  end for

end for

{**Phase II: Virtual Backbone based Convergecast Scheduling**}

for $k = 1; k \le h; k++$ **do**

  for $i = 1; i \le \kappa/\kappa'; i++$ **do**

    for $j = 1; j \le \kappa/\kappa'; j++$ **do**

      schedule all the cells in $\mathbb{S}_{i,j} \cap \mathcal{L}_k$ $(\mathbb{S}'_{i,j} \cap \mathcal{L}_k)$ simultaneously: $\forall \mathcal{C}_{u,v} \in \mathbb{S}_{i,j} \cap \mathcal{L}_k$ $(\mathbb{S}'_{i,j} \cap \mathcal{L}_k)$, $\mathfrak{B}(\mathcal{C}_{u,v})$ aggregates all the data packets it received and transmits an aggregation value to its parent SU along the convergecast tree $\mathcal{T}$

    end for

  end for

end for

In the following, we analyze the delay performance of CS. From Lemma 1, the average number of SUs in a cell is $3\log n$, which is referred as the *average case*, and the maximize number of SUs in a cell is $8\log n$, which is referred as the *worst case*. Then, the following Theorem 3 which shows the delay performance of CS can be obtained.

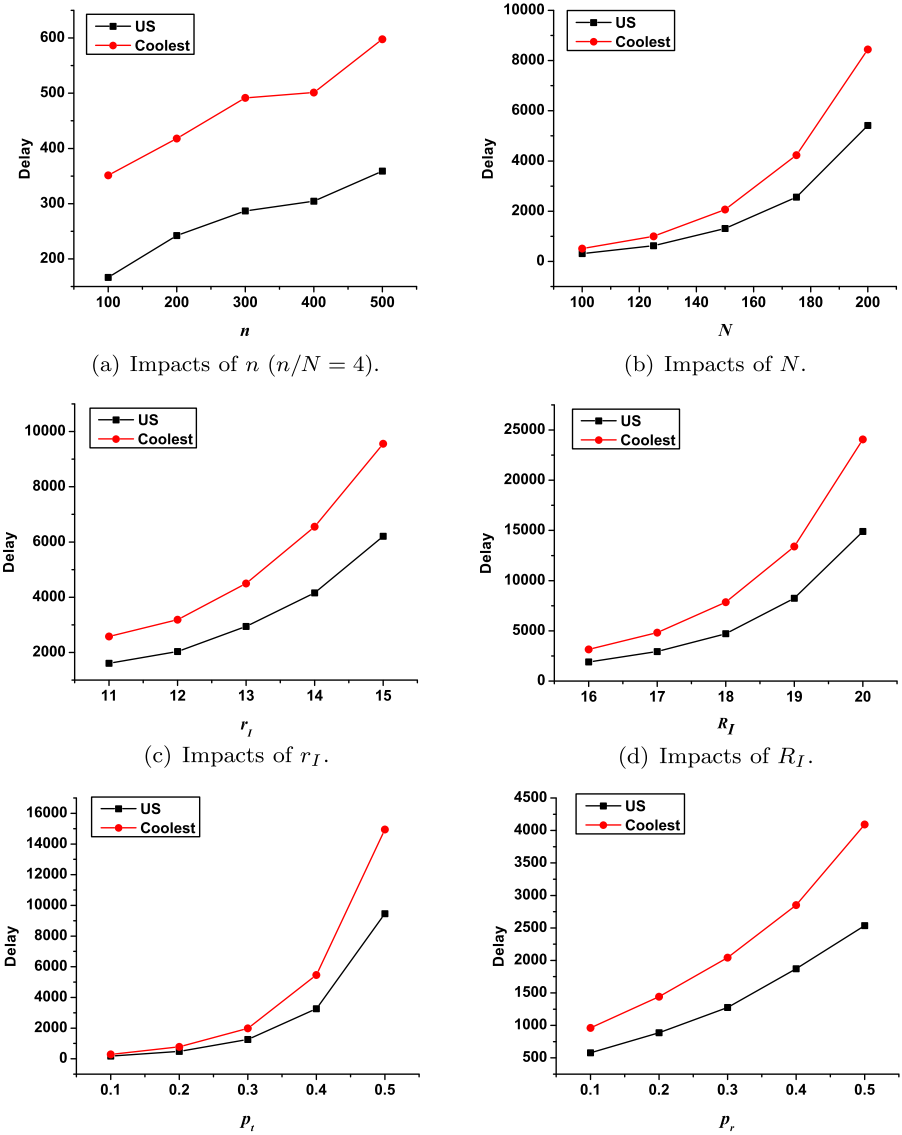
Theorem 3 (i) *When the convergecast is carried over physical links, the expected delay of CS is less than* $\frac{3\tau\kappa^2\log n}{p_p}+\frac{(2\kappa-1)h\tau}{p_p}=O(\log n+\sqrt{n/\log n})$ *in the average case and less than* $\frac{8\tau\kappa^2\log n}{p_p}+\frac{(2\kappa-1)h\tau}{p_p}=O(\log n+\sqrt{n/\log n})$ *in the worst case.* (ii) *Similarly, when the convergecast is carried over guaranteed physical links, the expected delay of CS is less than* $\frac{3\tau\kappa'^2\log n}{p_g}+\frac{(2\kappa'-1)h\tau}{p_g}=O(\log n+\sqrt{n/\log n})$ *in the average case and less than* $\frac{8\tau\kappa'^2\log n}{p_g}+\frac{(2\kappa'-1)h\tau}{p_g}=O(\log n+\sqrt{n/\log n})$ *in the worst case.*

Proof (i) In the first phase, all the cells in a concurrent cell set can conduct data aggregations concurrently. Within each cell, all the non backbone nodes transmit their data to the backbone node in that cell, which takes less than $\frac{3\tau\log n}{p_g}$ time in the average case and less than $\frac{8\tau\log n}{p_g}$ time in the worst case. We have κ^2 concurrent cell sets, which implies the expected time of the first phase is less than $\frac{3\tau\kappa^2\log n}{p_p}$ in the average case and less than $\frac{8\tau\kappa^2\log n}{p_p}$ in the worst case. In the second phase, all the backbone nodes convergecast their data to the sink level by level along the convergecast tree. In each level, the cells come from at most $2\kappa-1$ concurrent cell sets based on the definition of concurrent cell sets. Since the convergecast tree has h levels, it follows that the expected time consumption is at most $\frac{(2\kappa-1)h\tau}{p_p}$ in both the average case and the worst case. In summary, when the convergecast is carried over physical links, the expected delay of CS is less than $\frac{3\tau\kappa^2\log n}{p_p}+\frac{(2\kappa-1)h\tau}{p_p}=O(\log n+\sqrt{n/\log n})$ in the average case and less than $\frac{8\tau\kappa^2\log n}{p_p}+\frac{(2\kappa-1)h\tau}{p_p}=O(\log n+\sqrt{n/\log n})$ in the worst case.

(ii) By a similar method in (i), the conclusions in (ii) are true. □

6 Simulation and analysis

In this section, we examine the performance of the proposed schemes via simulations. For the system parameters, we denote them by the same symbols as before. For instance, n/N is the number of secondary/primary users, r/R is the transmission radius of a SU/PU, r_I/R_I is the interference radius of a SU/PU, p_t/p_r is the probability that a PU becomes to a transmitter/receiver during a time slot. Moreover, we define $\rho=r^2\cdot n/A$ as the *communication density* of the SUs. The proposed scheme is compared with *Coolest* routing algorithm (Huang et al. 2011), which is published recently. In Coolest, the path with the most balanced and/or the lowest spectrum utilization by the PUs is preferred for a data transmission. We modify the Coolest routing algorithm to form a data aggregation tree by determining a path from each



(a) Impacts of n ($n/N = 4$). (b) Impacts of N.

(c) Impacts of r_I. (d) Impacts of R_I.

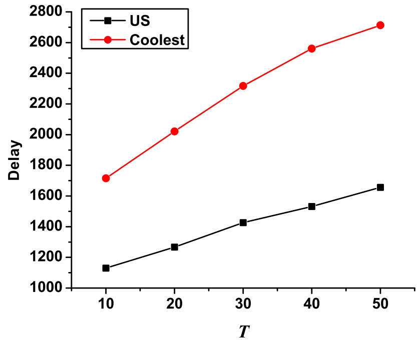
(e) Impacts of p_t. (f) Impacts of p_r.

Fig. 7 Performances of US and Coolest under different system parameters. Without specification, the default settings are $\rho = 4$, $n = 400$, $N = 150$, $r = 10$, $r_I = 10$, $R = 15$, $R_I = 15$, $p_t = 0.3$, $p_r = 0.3$, and $T = 20$

SU to the sink. The proposed convergecast scheduling scheme is compared with a modified version of Coolest. The value reported in simulations is the average value across 100 runs.

Figure 7 demonstrates the unicast performance between US and Coolest. Suppose there are T unicast tasks in the secondary network. Figure 7(a)–(g) show

Fig. 7 (*Continued*)



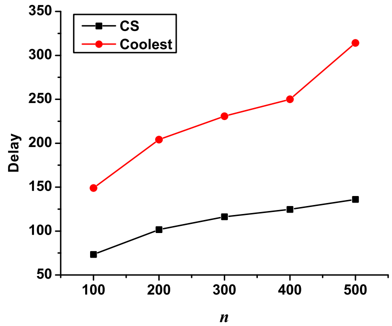
(g) Impacts of T.

that the delays of US and Coolest increase with any individual parameter from $\{n, N, r_I, R_I, p_t, p_r, T\}$ increase. US outperforms Coolest in all the cases. To be specific, US induces 73.58%, 59%, 55.65%, 63%, 60.17%, 59.35% and 61.54% less delay compared with Coolest on average for parameters n, N, r_I, R_I, p_t, p_r and T, respectively. The reason that US has a better performance than Coolest is because: (i) in Coolest, the path with the lowest spectrum utilization by the PUs is preferred for a data transmission. Therefore, the path with more spectrum opportunities will become crowded in Coolest because will schedule $T = 20$ unicast tasks simultaneously and these unicast tasks preferred the common links with more spectrum opportunities; (ii) in US, we conduct unicast transmissions over the shortest path through the cell-based virtual backbone. When multiple unicast tasks are scheduled simultaneously, the cells from a (guaranteed) concurrent cell set can transmit data concurrently according to Corollary 1. This accelerates the data transmission process of US.

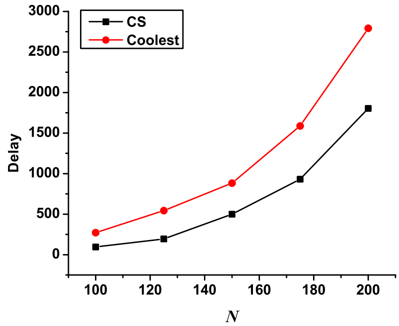
Figure 8 shows the performance comparison of convergecast. We randomly choose one backbone node as the sink and all the data of a snapshot will be aggregated to the sink. Figure 8(a)–(f) show that the delays of US and Coolest increase with the any individual parameter from $\{n, N, r_I, R_I, p_t, p_r\}$ increase. Again, US outperforms Coolest in all the cases. US induces 108.01%, 72.45%, 48.32%, 58%, 51.65% and 67.59% less delay compared with Coolest on average for parameters n, N, r_I, R_I, p_t, and p_r, respectively.

7 Conclusion

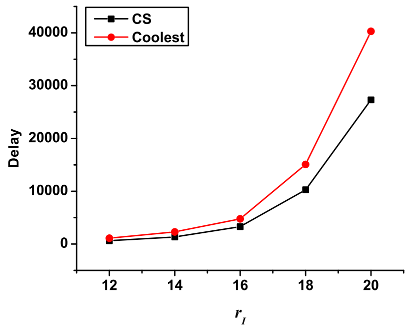
In this paper, we first construct a cell-based virtual backbone for a CRN. By theoretical analysis, we prove that the expected one hop delay in the constructed virtual backbone is upper bounded by a constant if the density of PUs is finite. Secondly, based on the constructed virtual backbone, we propose a *Unicast Scheduling* (US) algorithm. The induced delay of US scales linearly with the distance between the data source and the data destination. Thirdly, we propose a two-phase virtual backbone based *Convergecast Scheduling* (CS) algorithm. Theoretical analysis of CS shows that its expected delay is upper bounded by $O(\log n + \sqrt{n/\log n})$. The simulation results also indicate that the proposed algorithms can finish the unicast and convergecast tasks effectively.



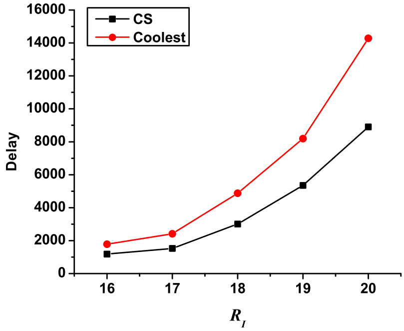
(a) Impacts of n ($n/N = 4$, $r_I = 10$, $R_I = 15$).



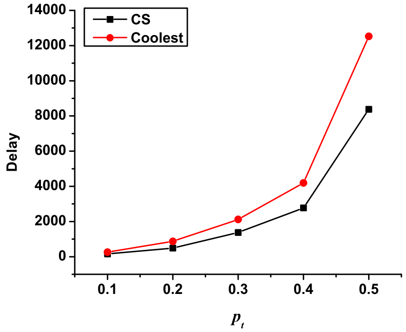
(b) Impacts of N ($r_I = 10$, $R_I = 15$).



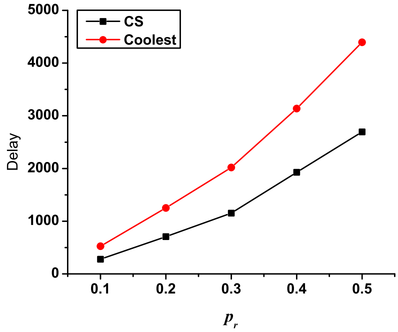
(c) Impacts of r_I ($R_I = 15$).



(d) Impacts of R_I ($r_I = 12$).



(e) Impacts of p_t ($r_I = 12$, $R_I = 16$).



(f) Impacts of p_r ($r_I = 12$, $R_I = 16$).

Fig. 8 Performances of CS and Coolest under different parameters. Without specification, the default settings are $\rho = 4$, $n = 400$, $N = 150$, $r = 10$, $R = 15$, $p_t = 0.3$, and $p_r = 0.3$

References

Akyildiz IF, Lee WY, Vuran MC, Mohanty S (2006) NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey. Comput Netw 50:2127–2159

Dousse O, Baccelli F, Thiran P (2003) Impact of interferences on connectivity in ad hoc networks. In: INFOCOM 2003, pp 1724–1733

Han C, Yang Y (2011) Information propagation speed study in multihop cognitive radio networks. In: INFOCOM 2011, pp 226–230
He J, Cai Z, Ji S, Beyah R, Pan Y (2011) A genetic algorithm for constructing a reliable MCDS in probabilistic wireless networks. In: WASA 2011, pp 96–107
Hu D, Mao S, Reed JH (2009) On video multicast in cognitive radio networks. In: INFOCOM 2009, pp 2222–2230
Huang W, Wang X (2011) Throughput and delay scaling of general cognitive networks. In: INFOCOM 2011, pp 2210–2218
Huang S, Liu X, Ding Z (2009) Optimal transmission strategies for dynamic spectrum access in cognitive radio networks. IEEE Trans Mob Comput 8(12):1636–1648
Huang X, Lu D, Li P, Fang Y (2011) Coolest path: spectrum mobility aware routing metrics in cognitive ad hoc networks. In: ICDCS 2011, pp 182–191
Jeon SW, Devroye N, Vu M, Chung SY, Tarokh V (2008) Cognitive networks achieve throughput scaling of a homogeneous network. IEEE Trans Inf Theory 57(8):5103–5115
Jin J, Xu H, Li B (2010) Multicast scheduling with cooperation and network coding in cognitive radio networks. In: INFOCOM 2010, pp 1–9
Kasbekar GS, Sarkar S (2010) Spectrum pricing games with bandwidth uncertainty and spatial reuse in cognitive radio networks. In: MobiHoc 2010, pp 251–260
Kim H, Shin KG (2008) In-band spectrum sensing in cognitive radio networks: energy detection or feature detection. In: MobiCom 2008, pp 14–25
Kompella S, Nguyen GD, Wieselthier JE, Ephremides A (2011) Stable throughput tradeoffs in cognitive shared channels with cooperative relaying. In: INFOCOM 2011, pp 1961–1969
Li C, Dai H (2011a) On the throughput scaling of cognitive radio ad hoc networks. In: INFOCOM 2011, pp 241–245
Li C, Dai H (2011b) Transport throughput of secondary networks in spectrum sharing systems. In: INFOCOM 2011, pp 2732–2740
Li Y, Thai MT, Wang F, Du DZ (2006) On the construction of a strongly connected broadcast arborescence with bounded transmission delay. IEEE Trans Mob Comput 5(10):1460–1470
Pan M, Zhang C, Li P, Fang Y (2011) Joint routing and link scheduling for cognitive radio networks under uncertain spectrum supply. In: INFOCOM 2011, pp 2237–2245
Ren W, Zhao Q, Swami A (2010) On the connectivity and multihop delay of ad hoc cognitive radio networks. In: ICC 2010, pp 1–6
Ross SM (2007) Introduction to probability models, 9th edn. Elsevier, New York
Shi Y, Hou YT (2008) A distributed optimization algorithm for multi-hop cognitive radio networks. In: INFOCOM 2008, pp 1292–1300
Shu T, Krunz M (2009) Coordinated channel access in cognitive radio networks: a multi-level spectrum opportunity perspective. In: INFOCOM 2009, pp 2976–2980
Shu T, Krunz M (2010) Truthful least-priced-path routing in opportunistic spectrum access networks. In: INFOCOM 2010, pp 1–9
Sun L, Wang W (2011) On distribution and limits of information dissemination latency and speed in mobile cognitive radio networks. In: INFOCOM 2011, pp 246–250
Wang F, Krunz M, Cui S (2008) Spectrum sharing in cognitive radio networks. In: INFOCOM 2008, pp 1885–1893
Wang C, Li XY, Tang S, Jiang C (2010) Multicast capacity scaling for cognitive networks: general extended primary network. In: MASS 2010, pp 262–271
Wang P, Akyildiz IF, Al-Dhelaan AM (2011) Percolation theory based connectivity and latency analysis of cognitive radio ad hoc networks. Wirel Netw 17:659–669
Yin C, Gao L, Cui S (2010) Scaling laws for overlaid wireless networks: a cognitive radio network versus a primary network. IEEE/ACM Trans Netw 18(4):1317–1329
Zhang T, Tsang DHK (2011) Optimal cooperative sensing scheduling for energy-efficient cognitive radio networks. In: INFOCOM 2011, pp 2723–2731