Le *et al. EURASIP Journal on Wireless Communications and Networking* (2015) 2015:49
DOI 10.1186/s13638-015-0289-2

EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Distributed back-pressure scheduling with opportunistic routing in cognitive radio networks

Yuan Le[1*], Xiuzhen Cheng[1], Dechang Chen[2], Nan Zhang[1], Taieb Znati[3], Mznah A Al-Rodhaan[4] and Abdullah Al-Dhelaan[4]

Abstract

A challenging problem in cognitive radio networks (CRNs) is to design a throughput efficient routing and scheduling algorithm for end-to-end communications between secondary users (SUs) in a distributed manner. Due to the opportunistic nature of CRN routing, we consider a framework with randomized path selection in this paper. Motivated by the back-pressure scheduling policy, we adopt the differential queue backlog as the routing metric to achieve throughput efficiency. Moreover, we demonstrate the sufficient condition for our framework to achieve throughput optimality by analyzing the Lyapunov drift. We further propose a distributed medium access control algorithm that can approximately satisfy the required condition when the transmission attempt probability is low. The performance of our proposed scheme is evaluated in a simulation platform, and the evaluation results verify the effectiveness of our distributed scheme.

Keywords: Cognitive radio networks; Opportunistic routing; Back-pressure scheduling; Throughput optimal routing

1 Introduction

Cognitive radio networking has been extensively studied in recent years [1-6]. The cognitive radio (CR) technique allows unlicensed users (secondary users (SUs)) to opportunistically access the licensed spectrum without interfering with licensed users (primary users (PUs)) to exploit the under-utilized portion of the spectrum [7-9]. The coexistence of PUs and SUs makes it infeasible to directly adopt the routing algorithms designed for general multi-hop wireless networks, as the links can be highly dynamic due to the existence of PUs. In the worst case, it may not be possible to deliver a packet along a certain path without interfering with any PU. With such a dynamic environment, designing a reliable routing protocol for a CR network (CRN) becomes a challenging problem.

There exist many routing algorithms that simultaneously consider spectrum sensing and management. For instance, a spectrum-aware routing algorithm was developed in [10] to integrate spectrum discovery with route establishment and then assign channels to links to minimize interference. But because the network topology in a CRN varies over time due to the dynamic nature of PU activities, many existing routing algorithms are rendered inapplicable as their computed paths usually do not adapt to the network dynamism. Thus whenever the network topology changes, routes may need to be recalculated, leading to a high computational cost. To address this challenge, opportunistic routing (OR) is introduced to CRN, in which SUs select only the next hop relay for their transmissions based on the routing metric computed from the current link states. For instance, a delay efficient opportunistic routing algorithm is proposed in [11].

On the other hand, the back-pressure (BP) framework proposed by Tassiulas et al. [12] introduces throughput optimal routing and scheduling to support multiple flows in wireless *ad hoc* networks. Observing that OR provides an opportunity to overcome the challenges caused by network dynamics (OR selects the next relay for routing based on the current link quality) and that BP scheduling has the potential to guarantee throughput

*Correspondence: ppzt@gwmail.gwu.edu
[1] Department of Computer Science, The George Washington University, Washington, DC, USA
Full list of author information is available at the end of the article

optimality (BP selects the flow with the highest back pressure), we decide to combine the concepts of back pressure and opportunistic routing, aiming at achieving throughput optimal routing in dynamic multi-hop cognitive radio networks. Furthermore, to avoid the centralized link controller in BP, we adopt CSMA/CA for collision avoidance among all selected flows. To the best of our knowledge, we are the first to combine BP and OR to take advantage of the benefits brought by OR and BP in a distributed fashion. Because the general back-pressure framework consists of a max-weight optimization problem, which is NP complete, we have to seek a greedy algorithm to implement our framework of distributed BP scheduling with OR routing in CRNs for a suboptimal solution.

In this paper, we adopt the queue model employed by the back-pressure algorithm in [12] to combine OR with BP and employ the differential queue backlog as our routing metric. We further analyze the Lyapunov drift of our framework and derive the sufficient condition to achieve throughput optimality. Leveraging this condition, we propose a distributed medium access control (MAC) algorithm that stabilizes the network when the traffic load is light. The core idea of our algorithm is to tune the contention window size (CW) corresponding to the maximum queue backlog to achieve the required transmission probability. We evaluate our protocol in a simulation platform, and our results indicate that the proposed protocol can achieve both throughput optimality and reliability under a light traffic load.

The rest of the paper is organized as follows. Section 2 discusses the most related work. The distributed back-pressure scheduling with opportunistic routing framework, together with its throughput optimality analysis, are presented in Section 3. We propose a distributed MAC algorithm that can stabilize the network when the transmission probability is low in Section 4. The evaluation settings and simulation results are reported in Section 5. We conclude this paper with a discussion on our future research in Section 6.

2 Related work

There exist many studies on the routing protocol design for cognitive radio networks [10,13-17]. A high-throughput spectrum-aware routing protocol was investigated in [10]. In [13], the authors developed a tree-based routing algorithm that can achieve low end-to-end delay. These protocols all use a predefined single path for all transmissions. Since the connectivity of a link may be highly dynamic in a cognitive radio network, the performance can be significantly decreased if a crucial link is frequently unavailable. To address this problem, a multi-path-based routing algorithm was proposed in [14]. To establish a more reliable end-to-end path, an opportunistic routing framework was adopted in [15]. To enhance throughput and obtain a global-path stable routing path, a network formation game was introduced to model the process of multi-hop relay selection in a cooperative CRN [16]. Mobility-assisted routing to deal with the intermittently connected characteristic of mobile *ad hoc* CRNs was investigated in [17].

Opportunistic routing has also been investigated in wireless mesh networks (WMNs) [11,18-23]. The widely studied ExOR routing protocol was proposed by Biswas et al. in [18]. ExOR selects the next hop relay node after the transmission is done. It adopts sequential acknowledgements to exchange priority information among all the receivers when a transmission occurs and selects the receiver with the highest priority as the next hop relay to forward the packet towards its destination. In contrast, Chachulski et al. developed a MAC-independent opportunistic routing scheme named MORE in [19], which combines routing with network coding to avoid node coordination. A contention-aware delay analysis of opportunistic routing was presented in [20], which provides a mathematical model about the contention to compute the expected delay. The inherent properties of information dissemination in opportunistic networks was studied in [21], and [22] proposes two algorithms for intermittently connected mobile P2P networks, which exploit the spatial locality, spatial regularity, and activity heterogeneity of human mobility to select relays.

The original back-pressure algorithm proposed by Tassiulas et al. in [12] has become an appealing research area due to its guarantee of throughput optimality. However, it involves a max-weight optimization problem, making the algorithm computationally inefficient. To address this challenge, many suboptimal solutions have been developed, which can be generally categorized into two classes. The first class [24,25] requires a centralized controller to make the scheduling decisions and focuses on developing scheduling algorithms whose stability can be guaranteed in a fraction of the capacity region. The impact of an imperfect scheduling policy on the capacity region was demonstrated by Lin et al. in [24]. Dimakis et al. [25] indicated that the full capacity region can be achieved by a greedy scheduling policy for certain networks. The second class algorithms [26-28] study distributed scheduling policies. In [26], the authors considered a datagram network and developed a distributed scheduling policy using only local information. Neely et al. [27] proposed a distributed approach using randomized transmission selection. In [28], a distributed greedy protocol was presented to achieve stability in a reduced capacity region. A learning-based distributed back-pressure scheduling algorithm requiring no information exchange was proposed in [29]; but it requires a relatively long convergence time.

In addition to finding suboptimal solutions for the original max-weight optimization problem, utility optimization based protocol design using the back-pressure framework has also being extensively studied. In [30], Georgiadis et al. claimed that the back-pressure algorithm performs within $O(1/V)$ of optimal throughput utility, with $O(V)$ average delay, where V is a control parameter that can be chosen as large as desired. This gives an explicit $[\,O(1/V), O(V)]$ utility-delay tradeoff. With different definitions of the utility functions, [31-33] proposed different back-pressure policies that may jointly consider power control, fairness, end-to-end delay, and flow control.

In this paper, we propose a distributed framework that combines opportunistic routing with back pressure scheduling and derive the sufficient condition for our framework to achieve throughput optimality. Based on this condition, we design a MAC protocol that stabilizes the network when the transmission probabilities are low. To our best knowledge, this paper is the first to take advantage of the benefits provided by BP and OR to tackle the challenges faced by cognitive radio networks with dynamic link statuses.

3 A framework of distributed back-pressure scheduling with opportunistic routing

3.1 Our distributed scheduling framework

In this section, we detail our framework of distributed back-pressure scheduling with opportunistic routing for a CRN with multiple simultaneous flows. Back-pressure scheduling is introduced to select the flow with maximum back pressure at each node while opportunistic routing is introduced to address the high dynamism of CRNs, such that the next hop relay for a flow can be selected opportunistically from all the feasible paths. Since the original back-pressure algorithm assumes the existence of a global controller that prevents the simultaneous activation of potentially conflicting links/flows for collision avoidance, we adopt CSMA/CA in our distributed framework for medium access control (MAC) as it is not feasible to implement a centralized link controller in a CRN. In summary, our distributed framework takes back-pressure scheduling for flow selection, opportunistic routing for path determination, and CSMA/CA for MAC regulation.

We consider a multi-hop CR network with N nodes and L links. Assume that there exist F flows in the network. For simplicity, we further assume that there are F flow queues at each node, with the cth queue serving the cth flow. The exogenous packet arrival rate at node i for an arbitrary flow c is denoted by a_i^c, and the corresponding arrival rate vector is denoted by A^c. Denote by $X_i^c(t)$ the queue length of flow c in node i. Let $s^c(l)$ and $d^c(l)$ be the sending and receiving nodes of link l for flow c, respectively. The connectivity and routing matrix $R^c = \left\{r_{il}^c\right\}$ for flow c is defined by a $N \times L$ matrix, with r_{il}^c representing the lth element in the ith row of R^c:

$$r_{il}^c = \begin{cases} 1, & \text{if } i = d^c(l), \\ -1, & \text{if } i = s^c(l), \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Note that the link connectivity in a CR network may vary over time and thus cause the R^c matrix to change.

Let p_l^c be the probability that link l is used to transmit packets for flow c. Thus, the queue length in time slot $(t+1)$ can be represented as follows:

$$X_i^c(t+1) = X_i^c(t) + \sum_{l=1}^{L} r_{il}^c \cdot b_l \cdot p_l^c + a_i^c, \tag{2}$$

where a_i^c is the arrival rate of flow c at node i and b_l is the transmission rate of link l.

Let P_i be the probability that node i can successfully transmit a packet. By adopting opportunistic routing, a packet can be successfully transmitted if at least one of its feasible relays can receive the packet correctly. Thus, P_i can be represented as

$$P_i = \sum_{c} \sum_{l:s^c(l)=i} p_l^c. \tag{3}$$

Let B be a diagonal matrix in which the lth diagonal element is b_l. Then, the equivalent matrix expression of $X^c(t+1)$ becomes:

$$X^c(t+1) = X^c(t) + R^c B P^c + A^c. \tag{4}$$

A link backlog D_l for each link is defined as follows. The original back-pressure routing policy [12] adopts the same definition.

$$D_l^c = b_l \cdot \left(X_{s^c(l)} - X_{d^c(l)}\right), \tag{5}$$

$$D_l = \max_c D_l^c \tag{6}$$

The best candidate flow and the corresponding backlog for node i, denoted by c_i^* and D_i^*, respectively, are selected according to the following policy:

$$c_i^* = \arg_c \max_{l:s^c(l)=i} D_l^c, \tag{7}$$

$$D_i^* = \max_{l:s^c(l)=i} D_l^c, \tag{8}$$

That is, the maximum D_l is selected from all the outbound links of node i, and its corresponding flow is selected as the best candidate flow.

In the next subsection, we derive the sufficient condition for our framework to achieve throughput optimality.

3.2 Throughput optimality of the proposed framework

We first define the stability of a network as follows.

Definition A network is stable under a scheduling policy π, if the queue length $X_i^c(t)$ satisfies:

$$\lim_{t\to\infty} \sup \sum_{c=1}^{F} \sum_{i=1}^{N} X_i^c(t) < D, \tag{9}$$

where D is a constant that is dependent on the arrival rate vectors A^c and transmission rate b_l.

Let C_π be the set of arrival rates for which the network is stable under policy π. The stability region C of a network is defined as $C = \bigcup_\pi C_\pi$.

Definition A scheduling policy is *throughput optimal* if the network is stable for any arrival rate $a \in C$ under the policy.

In the following, we present a lemma to show that our distributed scheduling framework proposed in Section 3.1 is throughput optimal under a certain success transmission probability P_i.

Lemma 1. *In a CSMA/CA based CRN, a back-pressure scheduling with distributed routing framework is throughput optimal if the following equation holds for all the nodes:*

$$P_i = \left(1 - \boldsymbol{O}\left(\frac{1}{D_i^*}\right)\right). \tag{10}$$

where $\mathbf{O}(\cdot)$ *stands for the big-O notation.*

The detailed proof of Lemma 1 can be found in the Appendix.

Lemma 1 provides the sufficient condition for a CSMA/CA-based scheduling framework to achieve throughput optimality. In the next section, we propose a distributed approximately throughput optimal MAC algorithm based on Equation 10.

4 A distributed approximate MAC algorithm

In this section, we first derive the contention window size that can satisfy Equation 10. Then, a detailed protocol of our scheduling framework is proposed.

4.1 Contention window size derivation

In our previous section, we show that a CSMA/CA-based CRN is stable when its successful transmission probabilities $\{P_i\}$ satisfy a certain constraint. In a network with a centralized transmission controller, it is not hard to achieve the required successful transmission probabilities. However, in a wireless network where stations compete for medium access in a distributed manner, it is difficult to guarantee the transmission probabilities. In this subsection, we propose a distributed medium access control algorithm that approximately achieves the required transmission probabilities for throughput optimality.

In a CSMA/CA network, the transmission probability P_i of station i is determined by the attempt probabilities $\{P_a^i, i = 1 \ldots N\}$. We adopt the model in [34], in which

$$P_i = P_a^i * \prod_{j\neq i} \left(1 - P_a^j\right), \tag{11}$$

and the attempt probability can be expressed as the following function of the contention window size CW_i

$$P_a^i = \frac{2}{CW_i + 1}. \tag{12}$$

Notice that when the attempt probabilities approach to 0, the product $\prod_{j\neq i}\left(1 - P_a^j\right)$ converges to 1, leading to $P_i = P_a^i$. In a typical CRN, transmissions occur sparsely and opportunistically due to the existence of PUs. Therefore, we can approximately represent P_i by P_a^i. On the other hand, according to Lemma 1, the following equation should hold for every P_a^i and D_i^*:

$$P_a^i = 1 - \mathbf{O}\left(\frac{1}{D_i^*}\right). \tag{13}$$

This implies that when the maximum backlog D_i^* becomes larger, the attempt probability P_a^i is enlarged. Recall that P_i can be represented by P_a^i only when the attempt probabilities approach to 0. Hence, the maximum attempt probability should be bounded when the maximum backlog is increasing, so that the sparse transmission constraint can be guaranteed. Otherwise, high attempt probabilities could decrease the successful transmission probabilities and consequently cause congestions. And the congestions may significantly increase the queue, causing even higher attempt probabilities due to the retransmissions.

To address this problem, we limit the queue length to a finite number. When the queue length reaches its upper bound, all newly arriving packets are dropped. Let $Q_{\max}$ be the maximum queue length. Then, the maximum backlog $D_{\max}$ is equal to $Q_{\max}$. The maximum attempt probability is thus bounded by

$$P_a^{\max} = 1 - \mathbf{O}\left(\frac{1}{Q_{\max}}\right). \tag{14}$$

Given a $Q_{\max}$, we can have different values for $P_a^{\max}$ depending on $\mathbf{O}\left(\frac{1}{Q_{\max}}\right)$. To find a better $P_a^{\max}$, we consider the scenario where all queues in all stations reach $Q_{\max}$. In such a situation, an optimal $P_a^{\max}$ should maximize the total throughput among the network. In [34,35], it has been shown that in a saturated network, the aggregated throughput can be maximized by finding the optimal attempt probabilities. By adopting the analysis in [35], we can calculate the optimal attempt probability P_a^{opt} given the maximum collision duration T_c, the slot duration T_{slot}, and the number of stations N. The relationship between N and P_a^{opt} can be expressed as

$$P_a^{\mathrm{opt}} = \frac{\zeta_0}{N}, \tag{15}$$

where ζ_0 is the root of the equation

$$1 - \zeta = \left(1 - \frac{T_{\mathrm{slot}}}{T_c}\right) e^{-\zeta}.$$

For a given CRN, T_c and T_{slot} are predefined constants; thus, P_a^{opt} can be calculated by each station. By setting $P_a^{\max} = P_a^{\mathrm{opt}}$, we have

$$P_a^{\max} = 1 - \mathbf{O}\left(\frac{1}{D_{\max}}\right) = \frac{\zeta_0}{N}. \tag{16}$$

Let $\mathbf{O}\left(\frac{1}{D_i^*}\right)$ be a function of the form $\frac{c_1}{D_i^*+c_2}$. Notice that when the maximum backlog D_i^* is equal to 0, the station should not attempt to transmit. This can be expressed as:

$$\begin{cases} 1 - \dfrac{c_1}{0 + c_2} = 0 & (17) \\ 1 - \dfrac{c_1}{D_{\max} + c_2} = \dfrac{\zeta_0}{N}. & (18) \end{cases}$$

By solving the above equations, we have

$$c_1 = c_2 = \frac{D_{\max}(N - \zeta_0)}{\zeta_0}, \tag{19}$$

and thus the corresponding contention window size $CW\left(D_i^*\right)$ can be expressed as

$$CW\left(D_i^*\right) = 1 + 2 \cdot \frac{D_{max}(N - \zeta_0)}{\zeta_0 \cdot D_i^*}. \tag{20}$$

4.2 Approximation analysis

In this subsection, we show the approximation analysis of our algorithm, comparing to the optimal solution. As shown in Equation 11, the difference between the attempt probability P_a^i and the transmission probability P_i, denoted by the collision probability P_f^i can be expressed as:

$$P_f^i = P_a^i - P_i = P_a^i \cdot \left(1 - \prod_{j \neq i}\left(1 - P_a^j\right)\right). \tag{21}$$

Recall that the attempt probability is upper-bounded by the maximum attempt probability $P_a^{\max}$, thus we have

$$\begin{aligned} P_f^i &= P_a^i \cdot \left(1 - \prod_{j \neq i}\left(1 - P_a^j\right)\right) \\ &\leq P_a^{\max} \cdot \left(1 - \prod_{j \neq i}\left(1 - P_a^{\max}\right)\right) \\ &= P_a^{\max} \cdot \left(1 - \left(1 - P_a^{\max}\right)^{N-1}\right) \\ &= \frac{\zeta_0}{N}\left(1 - \left(1 - \frac{\zeta_0}{N}\right)^{N-1}\right). \end{aligned} \tag{22}$$

Considering the impact of a collision, it is trivial that a collision can lead to a retransmission. If we consider such additional transmissions as newly arrived packets, the equivalent arrival rate for these retransmissions can be expressed as $\sum_{i=1}^{N} P_f^i$. Thus, comparing to the ideal solution with zero collision probability, our proposed algorithm decreases the feasible arrival rate for an amount of $\sum_{i=1}^{N} P_f^i$. Hence, the difference between the maximum capacity region and the feasible capacity region of our proposed algorithm is no larger than

$$\sum_{i=1}^{N} P_f^i \leq N \cdot \frac{\zeta_0}{N}\left(1 - \left(1 - \frac{\zeta_0}{N}\right)^{N-1}\right). \tag{23}$$

4.3 Protocol description

Our protocol consists of four major steps: i) flow selection, ii) random backoff period, iii) transmission and acknowledgement exchange, and iv) forwarding decision. Similar to the original opportunistic routing protocol [36], we assume that a distance matrix is maintained by each station, which is used to calculate the distance to other stations.

When a station is ready to transmit, it first selects the flow with the maximum backlog. After the flow is selected, the CW value is calculated based on the maximum backlog, as described in Section 4.1, and a random backoff period is generated. The packet is then broadcast when the backoff period is passed. The distance between the station and the destination of the packet, as well as the current queue length of the flow, is appended to the packet header. Each station that receives the packet should compare the distance and queue length from the header with its own values. If the receiver is closer to the destination and the queue length of the corresponding flow is smaller than that of the sender, the receiver sends an acknowledgement back and considers itself as a candidate relay. Candidate relays decide whether or not to forward the received packet after all the acknowledgements are received. This procedure is repeated until the packet reaches the destination station.

4.3.1 Flow selection

In our protocol, each station selects the flow with the maximum backlog to serve. To calculate the backlog, each station maintains a matrix that includes the approximate queue length of its neighbors. When the station receives a packet or an acknowledgement message, the corresponding queue length value in the matrix is synchronized to the value included in the header.

4.3.2 Random backoff period

After a flow is selected, the maximum backlog D_i^* is used to calculate the CW size as defined in Equation 20. During the backoff period, the station continues monitoring the spectrum. If a packet transmission is sensed or a PU occupies the spectrum, the station should pause the backoff timer and wait until the channel becomes available again.

4.3.3 Transmission and acknowledgement

When the backoff period is passed, the station pops a packet from the flow queue and immediately starts transmitting. The distance between the current station and the destination station and the current queue length of the flow are both stored in the header. Then, the station waits for acknowledgement messages from its neighbors. Different from the default 802.11 MAC algorithm, we reserve multiple time slots for ACK exchanges in our protocol. In the *i*th slot, the recipient with the *i*th smallest queue length sends the ACK. This order can be calculated from the queue length matrix stored at each station. In the case where multiple stations have the same queue length, the stations send their ACKs in the order of their station IDs.

To indicate that a packet has been received successfully, the ACK also includes the station ID and the queue length of the best candidate relay known to the ACK sender. For instance, assume the queue length values of the flow *c* in stations *A*, *B*, and *C* are 2, 4, and 5, respectively. In this case, station *A* sends the ACK message first with its own ID and queue length value. In the second time slot, station *B* sends its ACK, reporting that station *A*, with queue length value 2, is the best candidate relay. In the last time slot, station *C* sends its ACK carrying the same best candidate relay information. This acknowledgement mechanism can prevent duplicate forwarding when an ACK message from a higher priority recipient is not received by a lower priority one. As shown in Figure 1, assume that the ACK message from station *A* is not received by station *B* due to the hidden terminal problem. At the second time slot, station *B* considers itself as the best candidate and sends an ACK message reporting *B* as the best candidate with queue length 4 as the minimum queue length. Station *C* receives the two ACKs and compares the two queue lengths. The smaller one, which is from station *A* in this example, is chosen as the best candidate and is broadcasted with the ACK message. Finally, station *B* receives the ACK from station *C* and updates its best candidate information.

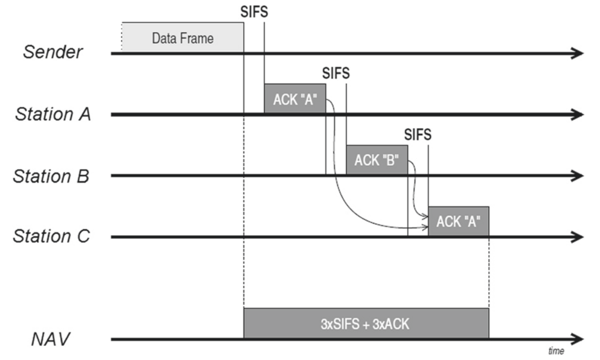

Figure 1 Illustration of sequential ACKs.

4.3.4 Forwarding decision

After exchanging the ACK messages, the stations make the decision regarding whether or not to forward the received packet. Specifically, after all ACK messages are received, if a station does not receive any ACK message with a smaller queue length value than itself, it pushes the received packet into the corresponding queue and later forwards it toward its destination. Otherwise, the station discards the received packet.

5 Performance evaluation

In this section, we evaluate the performance of our proposed transmission protocol using the *OMNET++* platform [37]. The average queue length, packet loss rate, throughput, and end-to-end delay are all evaluated in our simulation tests.

5.1 Simulation settings

We adopt the *INETMANET* framework in OMNET++ [37] as our cornerstone. The MAC and IP modules are modified to implement our transmission protocol. In our simulation, the parameters of the MAC and PHY layer adopt the default values same as those in the 802.11b standard. Additionally, the transmit power and signal-to-interference-plus-noise ratio (SINR) thresholds are defined so that the transmission range is equal to 250 m. Table 1 lists the basic parameters employed in our simulation.

In our network, multiple stations are placed in the simulation area and set to be static. We assume the stations always transmit packets using the maximum transmission rate, which implies that rate adaptation is disabled.

Table 1 The 802.11b parameters in Omnet++

Slot time (μs)	SIFS (μs)	DIFS (μs)
20	10	50
ζ_0	MTU (bytes)	ACK length (bits)
0.1622	1,500	112
IP header (bits)	MAC header (bits)	PHY header (bits)
160	272	192
Bitrate	Bitrate of ACK	Bitrate of PHY
11 Mbps	2 Mbps	1 Mbps
Transmit power (mW)	SINR threshold (dB)	Sensitivity (dBm)
2	4	−85

The flows are defined at the beginning of the simulation, and their arrival rates remain the same throughout the entire simulation period. Additionally, we disable RTS/CTS exchanges. The maximum queue length of each queue is set to 15 in our simulation.

5.2 Methodology

Our evaluation is conducted under two different network scenarios. We first consider a simple multi-hop network with four stations located at the four vertices of a square. The edge length of the square is set to 200 m; thus, the two diagonal vertices cannot communicate directly. We define two flows in this network: the first flow transmits packets from the upper-left station to the lower-right station; and the other flow goes from the lower-right station to the upper-left one. Each flow can have two paths from their source nodes to their destinations. To evaluate the impact of link availability on network performance, we consider a simple PU deploy model, in which each station can be interrupted by one and only one PU. In addition, each PU individually occupies the channel with a certain occupancy rate. Please notice that the random PU deploy model is not adopted in this scenario because all the four stations are close to each other, and thus a randomly deployed PU could simultaneously interfere with all the stations at a high probability. The impact of the packet arrival rate and that of the primary user occupancy rate[a] on the network performance are evaluated for this scenario. The network topology and the flows are illustrated in Figure 2.

In our second scenario, we randomly deploy N stations and five primary users in a 1,500 by 1,500 m simulation area. The interference range of the primary users is set to 300 m; thus when a primary user occupies the channel, the stations within its interference range cannot transmit or receive any packets. In addition, L flows are defined in the network by randomly selecting L pairs of sources and destinations from the N stations. In this scenario, we study the impact of the packet arrival rate A, the total number of stations N, and the total number of flows L, on the performance of our protocol.

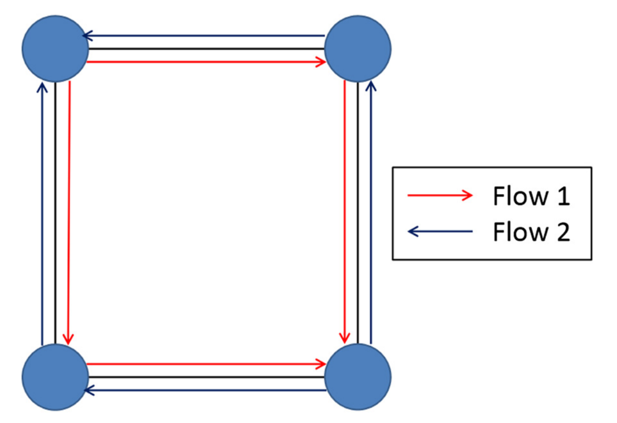

Figure 2 Network topology and the two flows for scenario I.

5.3 Evaluation results

The evaluation results are detailed in terms of the average queue length, the average end-to-end delay, the average packet loss rate, and the average throughput. All of our simulation results are the averaged value of ten runs.

5.3.1 Average queue length

Figure 3 shows the impact of the packet arrival rate on the average queue length in the simple network. The queue length calculated from Equations 10 and 19 is shown in the figure as well (labeled by *ideal queue length*). From this figure, we observe that the average queue length curve is comprised of three distinct phases. In the first phase, the time gap between the arrivals of two packets is longer than the length of the maximum backoff period; and thus,

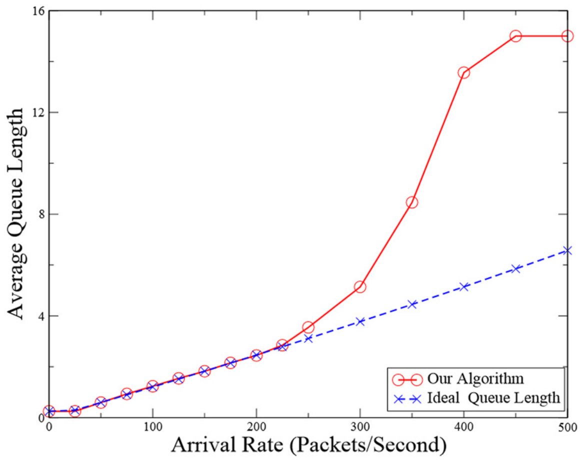

Figure 3 Average queue length vs. arrival rate.

the first packet is transmitted before the second packet arrives, although the backoff period of each packet is long. As a result, the average queue length remains 0 or 1 for the majority of the time. In the second phase, the average queue length grows linearly while the arrival rate increases. In this phase, packets reach their destinations at approximately the same rate as the arrival rate; thus, the transmission probability is proportional to the arrival rate. In the last phase, the low transmission attempt probability condition does not hold after the arrival rate exceeds some threshold. Accordingly, packets collide with others frequently, and the average queue length grows quickly to the upper bound. We also evaluate the variation of the average queue length for the second phase in Figure 4. From the figure, we observe that at the beginning, the queue length of the source station rapidly increases. The increasing queue length then reduces the CW size and thus increases the transmission probability. As a result, the queue length becomes stable after the transmission probability reaches the arrival rate.

In Figure 5, we evaluate the impact of the PU occupancy rate on the average queue length. In this evaluation, the arrival rate is set to 150 packets per second. We observe that the curve is similar to that of Figure 3. We can infer this result by considering the impact of PU occupancy as a reduction of the arrival interval. In other words, increasing the PU occupancy rate is approximately equivalent to increasing the packet arrival rate, although the increasing ratio may not necessarily be equal. Similar results are observed in the simulation of the other three metrics as well; thus, the impact of the PU occupancy rate will not be reported in the following subsections. Furthermore, the PU occupancy rate is set to 0.4 without further clarification.

The performance of our protocol in the large network is shown in Figures 6 and 7. From the figures, we can observe that the three phases exist in this scenario as well. In addition, when the arrival rate is low, the increased number of stations does not obviously impact on the performance, as the attempt probabilities of all the stations are very low, and thus the increasing number of stations does not obviously enlarge the collision probability. When the arrival rate grows, the impact of the amount of stations on the average queue length becomes more apparently. This is because the increasing number of stations lead to a higher collision probability and thus increase the queueing time for each packet. On the other hand, by increasing the number of flows, the threshold to transfer to phase three can be reached more quickly, and the average queue length grows faster. This result can be easily inferred as the aggregated arrival rate is proportional to the number of flows.

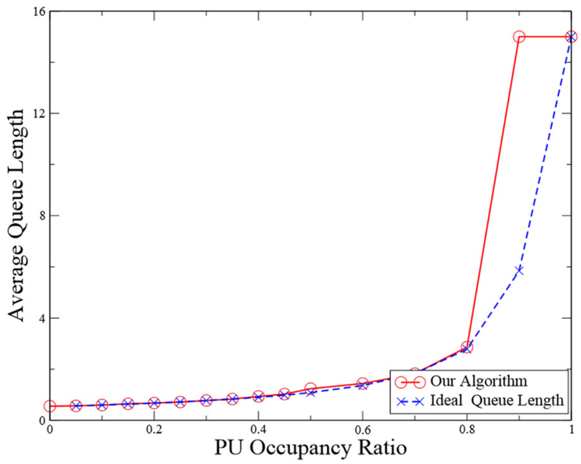

Figure 5 Average queue length vs. PU occupancy rate.

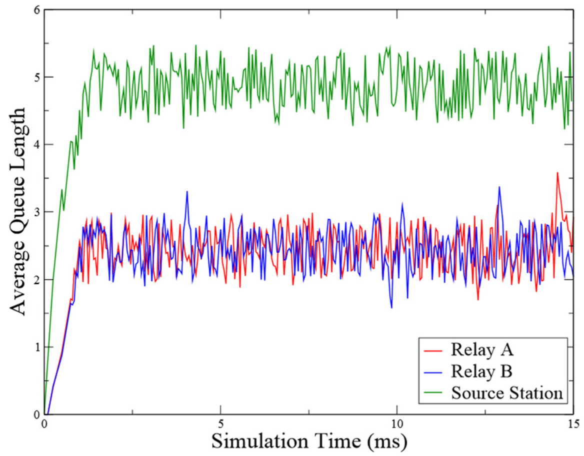

Figure 4 Average queue length vs. time.

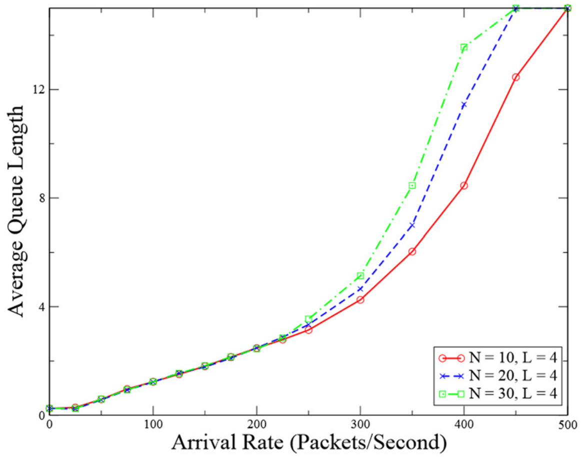

Figure 6 Average queue length vs. arrival rate with different number of stations.

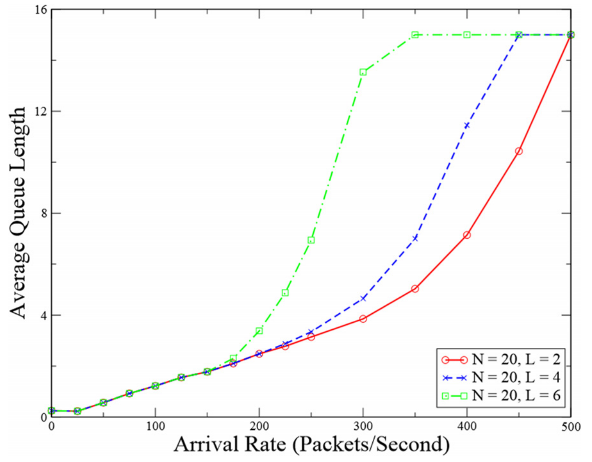

Figure 7 Average queue length vs. arrival rate with different number of flows.

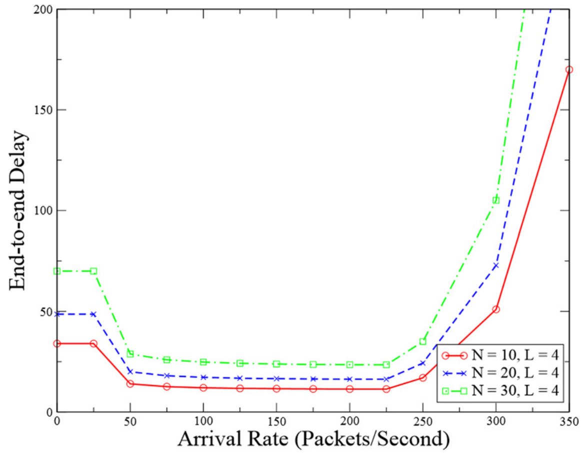

Figure 9 End-to-end delay vs. arrival rate with different number of stations.

5.3.2 End-to-end delay

We first illustrate the end-to-end delay in the simple scenario in Figure 8. From the figure, one can see that when the packet arrival rate is lower than 50 packets per second, the end-to-end delay is approximately equal to 20 ms. The major time cost in this case comes from the maximal backoff period. As the arrival rate increases, the end-to-end delay decreases rapidly, which is caused by the shortened backoff period corresponding to the increased queue backlog. Following this, the delay remains unchanged until the arrival rate exceeds the threshold for phase three. Once this threshold is reached, the end-to-end delay then converges to infinity. As the queue length exceeds the upper bound, packets start being dropped.

Figures 9 and 10 illustrate the end-to-end delay of our protocol in the second scenario. We notice that the end-to-end delay increases when N grows since the average number of hops for each flow is related to the network size, and the backoff period for each transmission is related to the number of stations within its transmission range. Also, the end-to-end delay increases when L becomes larger, as the queueing time in each hop is increased. Similar to the average queue length, the threshold to reach phase three becomes smaller while L grows.

5.3.3 Packet loss ratio

The packet loss ratio is measured by the ratio of the amount of lost packets to the total number of transmitted packets. Similar to the average queue length, we can observe the existence of a threshold and divide the curve into three phases. As shown in Figure 11, the packet loss ratio remains low until the threshold is reached. Then, the ratio increases rapidly to 1 as the entire network becomes congested; and consequently, most packets

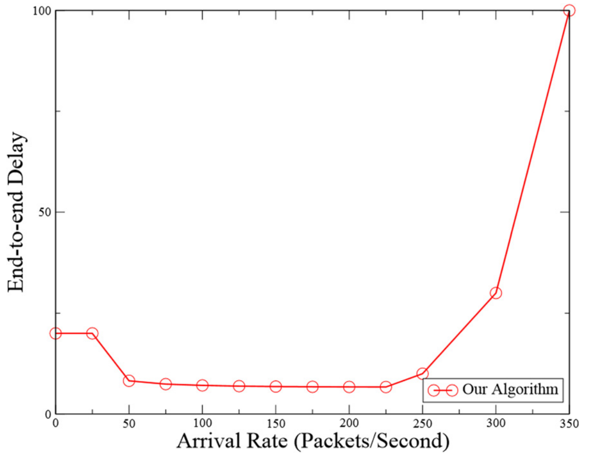

Figure 8 End-to-end delay vs. arrival rate.

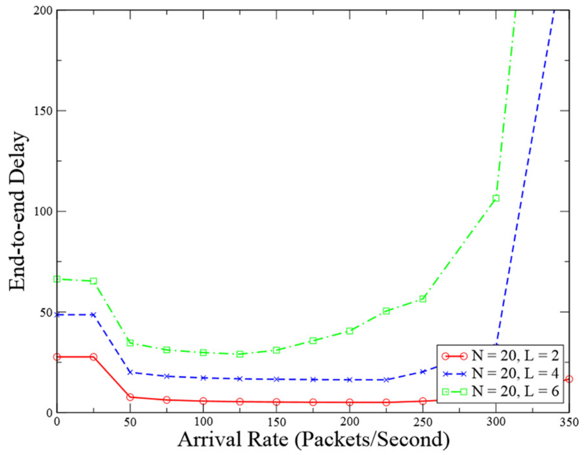

Figure 10 End-to-end delay vs. arrival rate with different number of flows.

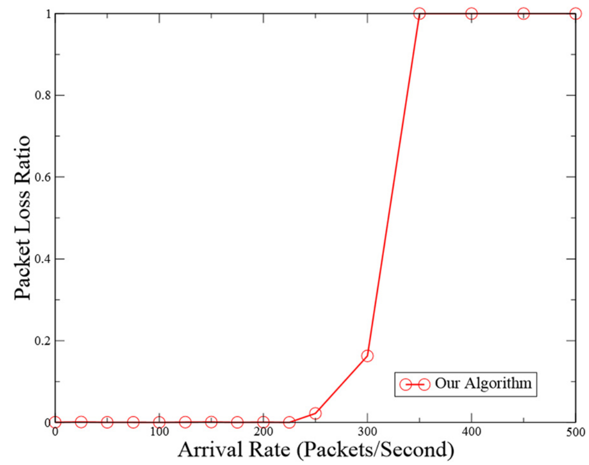

Figure 11 Packet loss ratio vs. arrival rate.

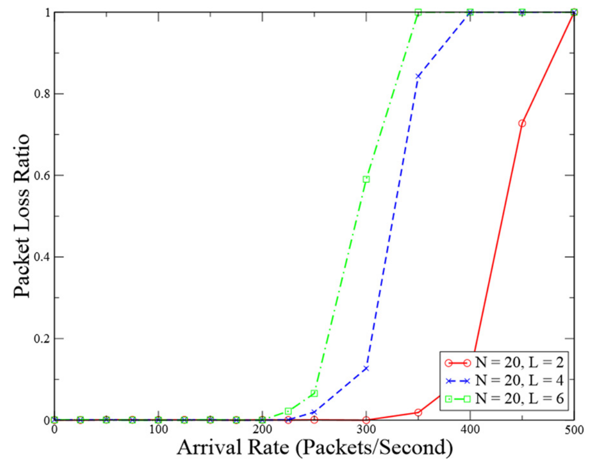

Figure 13 Packet loss ratio vs. arrival rate with different number of flows.

either are dropped by the stations or have collided with other packets.

The impacts of N and L on the packet loss ratio are reported in Figures 12 and 13, respectively. It can be observed that the impact of N and L on the packet loss ratio is in accordance to that on the average queue length.

5.3.4 Throughput

The throughput of the network can be measured as the total number of packets arrived at their destinations within a unit of time. The results are shown in Figures 14, 15, and 16. For the simple scenario, we notice that the throughput is equal to the aggregated arrival rate before the threshold is exceeded. After that, the throughput drops quickly to 0, as no packet can be delivered to the destination.

For the second scenario, from Figure 15, we can see that the thresholds are not affected by the total number of stations. When the arrival rate approaches to the threshold, the collision probability becomes higher, so some of the packets are dropped after multiple retransmission failure. A higher N can lead to a higher collision probability and thus can cause more packets being dropped. On the other hand, the increasing stations provide more queueing space for storing packets when the network is congested, as well as more feasible paths to the destination. So when the network is not fully congested, the performance of our algorithm does not have much difference when N increases. However, when the arrival rate becomes large enough, a network with a greater N is easier to be fully congested, as the collision probability grows faster than a network with a smaller N. In contrast, Figure 16 shows the

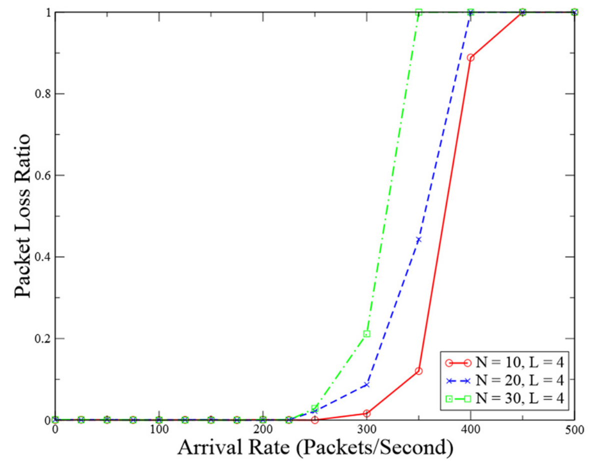

Figure 12 Packet loss ratio vs. arrival rate with different number of stations.

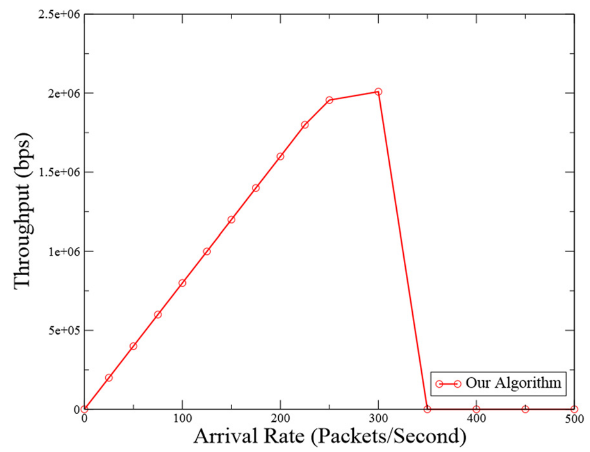

Figure 14 Throughput vs. arrival rate.

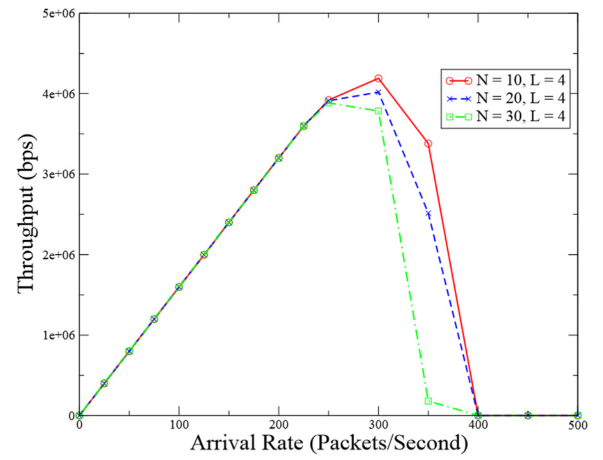

Figure 15 Throughput vs. arrival rate with different number of stations.

impact of L on the performance of our algorithm. It is clear that by increasing the number of flows, a higher aggregated throughput can be achieved when the arrival rate is low, because the aggregated arrival rate is proportional to the number of flows. At the same time, the network becomes much more congested, and thus the threshold is reached earlier.

5.3.5 Summary

The above evaluation results demonstrate that our proposed algorithm performs well when the arrival rate is smaller than a certain threshold. This threshold depends on the number of flows in the network but is insensitive to the network size.

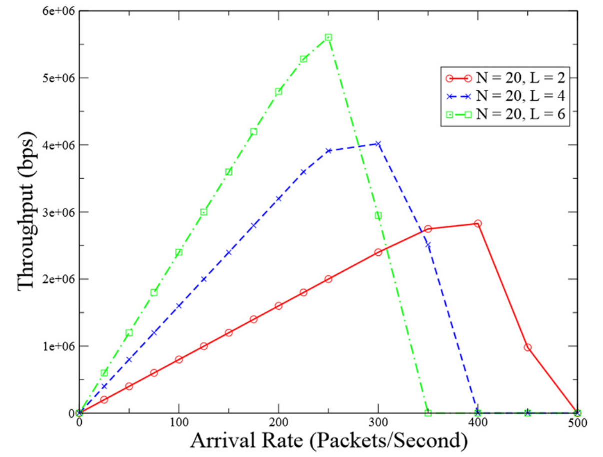

Figure 16 Throughput vs. arrival rate with different number of flows.

6 Conclusions

In this paper, we have proposed a framework of distributed back-pressure scheduling with opportunistic routing for CRNs and studied the sufficient condition for the proposed framework to achieve throughput optimality. To the best of our knowledge, we are the first to combine opportunistic routing with back-pressure scheduling in a distributed fashion. We also have proposed a novel distributed routing and medium access control protocol for end-to-end communications in a CRN. We have evaluated the performance of our protocol in a simulation-based study, and our evaluation results demonstrate that the proposed protocol is able to achieve both throughput optimality and reliability with the existence of PUs. As part of our future work, we plan to improve the end-to-end delay of our proposed protocol.

Endnote

[a] The PU occupancy rate refers to the probability of a PU occupying its channel at each time slot.

Appendix

[Proof of Lemma 1]

Proof Similar to the proof in [12], we consider a Lyapunov function

$$V(X(t)) = \sum_{c=1}^{F}\sum_{i=1}^{N}(X_i^c(t))^2.$$

To prove the throughput optimality, we need to show that the following relationships hold:

$$E\left[V(X(t+1)) - V(X(t))|X(t)\right] < \infty, \forall X(t), \qquad (24)$$

and for any given $\epsilon > 0$ and arrival rate $a \in C$, there exists a positive number b, such that

$$E\left[V(X(t+1)) - V(X(t))|X(t)\right] < -\epsilon, \text{for } \forall V(X(t)) > b \qquad (25)$$

In the following, we first prove that the inequality (Equation 24) holds.

$$\begin{aligned}
&E\left[V(X(t+1)) - V(X(t))|X(t)\right]\\
&= E\left[\sum_{c=1}^{F}\sum_{i=1}^{N}\left(\left(X_i^c(t+1)\right)^2 - \left(X_i^c(t)\right)^2\right)|X(t)\right]\\
&= E\left[\sum_{c=1}^{F}\sum_{i=1}^{N}\left(X_i^c(t+1) - X_i^c(t)\right)^2 |X(t)\right] \qquad (*)\\
&\quad + 2\cdot E\left[\sum_{c=1}^{F}\sum_{i=1}^{N}\left(X_i^c(t+1) - X_i^c(t)\right)\cdot X_i^c(t)|X(t)\right] \qquad (**)
\end{aligned}$$

For part ($**$), it is trivial to show that the amount of data directed in or out of a queue is bounded by the total capacity of the links connected to the queue. Thus, part $**$ is bounded by a constant, say, b_1.

Considering part ($*$), we have

$$
\begin{aligned}
&E\left[\sum_{c=1}^{F}\sum_{i=1}^{N}\left(X_i^c(t+1)-X_i^c(t)\right)\cdot X_i^c(t)|X(t)\right]\\
&=E\left[\sum_{c=1}^{F}\sum_{i=1}^{N}\left(\sum_{l=1}^{L} r_{il}^c\cdot b_l\cdot p_l^c+a_i^c\right)\cdot X_i^c(t)|X(t)\right]\\
&=E\left[\sum_{c=1}^{F}\sum_{i=1}^{N}\sum_{l=1}^{L} X_i^c(t)\cdot r_{il}^c\cdot b_l\cdot p_l^c|X(t)\right]\\
&\quad+E\left[\sum_{c=1}^{F}\sum_{i=1}^{N} X_i^c(t)\cdot a_i^c|X(t)\right]\\
&=-\sum_{c=1}^{F}\sum_{l=1}^{L} D_l^c(t+1)p_l^c+\sum_{c=1}^{F}(X^c)^T A^c
\end{aligned}
\tag{26}
$$

Clearly, the term $-\sum_{c=1}^{F}\sum_{l=1}^{L}D_l^c(t+1)P_l^c$ is bounded by 0. The proof in [12] indicates that $\sum_{c=1}^{F}(X^c)^T A^c$ is bounded by $D(t+1)^T\sum_{i=1}^{|S|}\lambda_i s_i$. Here, $S=\{s_i, i=1,2,\ldots,|S|\}$ is the set of all feasible activation vectors, and $\{\lambda_i\}$ is a set of positive factors that satisfy $\sum_{i=1}^{|S|}\lambda_i<1$. Thus, we conclude that inequality (24) holds for every $X(t)$.

In the following, we show that there exists a positive value b such that the inequality (Equation 25) holds when $V(X(t))>b$. In the previous proof, we have shown that the following inequalities hold:

$$
\sum_{c=1}^{F}\left(X^c\right)^T A^c\leq D(t+1)^T\sum_{i=1}^{|S|}\lambda_i s_i,
\tag{27}
$$

$$
E\left[\sum_{c=1}^{F}\sum_{i=1}^{N}\left(\left(X_i^c(t+1)\right)^2-\left(X_i^c(t)\right)^2\right)|X(t)\right]<b_1
\tag{28}
$$

Assume that there exists an activation vector $\hat{s}$ that can maximize $D(t+1)^T s$. Let $\hat{s}_l^c$ be the indicator for flow c on link l. Then, equation $\hat{s}_l^c=1$ holds if c is the corresponding flow to D_l and $\hat{s}_l$ is equal to 1. Considering the difference between $\sum_{c=1}^{F}\sum_{l=1}^{L}D_l^c(t+1)p_l^c$ and $D(t+1)^T\hat{s}$, we have:

$$
\begin{aligned}
&D(t+1)^T\hat{s}-\sum_{c=1}^{F}\sum_{l=1}^{L}D_l^c p_l^c\\
&=\sum_{c}\sum_{l}\hat{s}_l^c D_l^c-\sum_{l}\sum_{c}D_l^c p_l^c\\
&\leq\sum_{c}\sum_{l}D_l\left(\hat{s}_l^c-p_l^c\right)\\
&=\sum_{l}D_l\sum_{c}\left(\hat{s}_l^c-p_l^c\right)
\end{aligned}
\tag{29}
$$

If we represent a link l by its source and destination node pair, we can rewrite $\sum_l D_l\sum_c\left(\hat{s}_l^c-p_l^c\right)$ as:

$$
\sum_{l}D_l\sum_{c}\left(\hat{s}_l^c-p_l^c\right)=\sum_{i=1}^{N}\sum_{j\in N(i)}D_{ij}\sum_{c}\left(\hat{s}_{ij}^c-p_{ij}^c\right),
\tag{30}
$$

where $N(i)$ is the set of neighboring nodes of i. Then, we have:

$$
\begin{aligned}
&D(t+1)^T\hat{s}-\sum_{c=1}^{F}\sum_{l=1}^{L}D_l^c(t+1)p_l^c\\
&\leq\sum_{i=1}^{N}\sum_{j\in N(i)}D_{ij}\sum_{c}\left(\hat{s}_{ij}^c-p_{ij}^c\right)\\
&\leq\sum_{i}\sum_{j}D_i^*\sum_{c}\left(\hat{s}_{ij}^c-p_{ij}^c\right)\\
&=\sum_{i}D_i^*\sum_{j}\sum_{c}\left(\hat{s}_{ij}^c-p_{ij}^c\right).
\end{aligned}
\tag{31}
$$

Notice that for any node i, the total number of links that can transmit simultaneously is no more than 1. Thus, we have

$$
\sum_{j}\sum_{c}\hat{s}_{ij}^c\leq 1.
\tag{32}
$$

From Equations 31 and 32, we obtain:

$$
\begin{aligned}
&D(t+1)^T\hat{s}-\sum_{c=1}^{F}\sum_{l=1}^{L}D_l^c(t+1)p_l^c\\
&\leq\sum_{i}D_i^*\left(1-\sum_{j}\sum_{c}p_{ij}^c\right)\\
&=\sum_{i}D_i^*\left(1-P_i\right).
\end{aligned}
$$

By setting P_i to $\left(1-\mathbf{O}\left(\frac{1}{D_i^*}\right)\right)$, we have

$$
-\sum_{c=1}^{F}\sum_{l=1}^{L}D_l^c(t+1)p_l^c\leq -D(t+1)^T\hat{s}+C_0\cdot N,
\tag{33}
$$

where C_0 is a constant that is related to the function $\mathbf{O}\left(\frac{1}{D_i^s}\right)$. Then, with Equations 26, 27, 28, and 33, we have

$$
\begin{aligned}
&E\left[V(X(t+1)) - V(X(t))|X(t)\right]\\
&\leq b_1 + 2\left(-\sum_{c=1}^{F}\sum_{l=1}^{L} D_l^c(t+1)p_l^c + D(t+1)^T\sum_{i=1}^{|S|}\lambda_i s_i\right)\\
&\leq b_1 + 2\left(-D(t+1)^T\hat{s} + C_0 \cdot N\right) + 2D(t+1)^T\sum_{i=1}^{|S|}\lambda_i s_i\\
&\leq b_1 + 2C_0 \cdot N - 2D(t+1)^T\hat{s} + 2D(t+1)^T\hat{s}\sum_{i=1}^{|S|}\lambda_i\\
&\leq b_1 + 2C_0 \cdot N - 2D(t+1)^T\hat{s}\left(1 - \sum_{i=1}^{|S|}\lambda_i\right).
\end{aligned}
$$

As shown in [12], we have:

$$
\left(D(t+1)^T\hat{s}\right) \geq \frac{1}{N}\sqrt{\frac{b}{FN}} \cdot \min_l\{b_l\}, \quad \text{when } V(X(t)) > b \tag{34}
$$

Thus, when $V(X(t)) > b$, we have

$$
\begin{aligned}
&E\left[V(X(t+1)) - V(X(t))|X(t)\right]\\
&\leq b_1 + 2C_0 \cdot N - 2\left(\left(1 - \sum_{i=1}^{|S|}\lambda_i\right)\frac{1}{N}\sqrt{\frac{b}{FN}} \cdot \min_l\{b_l\}\right).
\end{aligned} \tag{35}
$$

By setting $b = NF \cdot \left[\frac{N(\epsilon + b_1 + 2C_0 \cdot N)}{2\left(1 - \sum_{i=1}^{|S|}\lambda_i\right)\min_l\{b_l\}}\right]^2$, we prove that the inequality (Equation 25) holds. Thus, the throughput optimal is achieved when Equation 10 holds for all the nodes.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work is supported in part by the National Science Foundation of the US under grants CNS-0831939, CNS-1162057, CNS-1162159, and CNS-1265311. The authors would also extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no. RGP-VPP-264.

Author details

[1]Department of Computer Science, The George Washington University, Washington, DC, USA. [2]Uniformed Services University of the Health Sciences, Bethesda, MD, USA. [3]Department of Computer Science, University of Pittsburg, Pittsburgh, PA, USA. [4]College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia.

Received: 29 August 2014 Accepted: 9 February 2015
Published online: 04 March 2015

References

1. M Song, C Xin, Y Zhao, X Cheng, Dynamic spectrum access: from cognitive radio to network radio. IEEE Wireless Commun. **19**(1), 23–29 (2012)
2. T Jing, X Chen, Y Huo, X Cheng, in *IEEE INFOCOM*. Achievable transmission capacity of cognitive mesh networks with different media access control (IEEE, New York, 2012), pp. 1764–1772
3. X Xing, T Jing, Y Huo, H Li, X Cheng, in *IEEE INFOCOM*. Channel quality prediction based on Bayesian inference in cognitive radio networks (IEEE, New York, 2013), pp. 1465–1473
4. X Xing, T Jing, W Cheng, Y Huo, X Cheng, Spectrum prediction in cognitive radio networks. IEEE Wireless Commun. **20**(2), 90–96 (2013)
5. Z Cai, S Ji, J He, AG Bourgeois, in *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference On*. Optimal distributed data collection for asynchronous cognitive radio networks (IEEE, New York, 2012), pp. 245–254
6. Z Cai, S Ji, J He, L Wei, AG Bourgeois, Distributed and asynchronous data collection in cognitive radio networks with fairness consideration. Parallel Distributed Syst. IEEE Trans. **25**(8), 2020–2029 (2014)
7. W Li, X Cheng, T Jing, Y Cui, K Xing, W Wang, Spectrum assignment and sharing for delay minimization in multi-hop multi-flow crns. IEEE J. Selected Areas Commun. (JSAC). **31**(11), 2483–2493 (2013)
8. W Li, X Cheng, RZhao Bie F, *An extensible and flexible truthful auction framework for heterogeneous spectrum markets* (ACM, New York, 2014), pp. 175–184
9. W Li, S Wang, X Cheng, R Bie, Truthful multi-attribute auction with discriminatory pricing in cognitive radio networks. ACM SIGMOBILE Mobile Comput. Commun. Rev. **18**(1), 3–13 (2014)
10. A Sampath, L Yang, L Cao, H Zheng, BY Zhao, in *Proc. of international conference on cognitive radio oriented wireless networks and communications (CROWNCOM)*. High throughput spectrum-aware routing for cognitive radio networks (IEEE, New York, 2007)
11. Z Cai, Y Duan, A Bourgeois, Delay efficient opportunistic routing in asynchronous multi-channel cognitive radio networks. J. Comb. Optimization, 1–21 (2013). doi:10.1007/s10878-013-9623-y
12. L Tassiulas, A Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. Automatic Control IEEE Trans. **37**(12), 1936–1948 (1992). doi:10.1109/9.182479
13. B Zhang, Y Takizawa, A Hasagawa, A Yamaguchi, S Obana, in *Wireless Communications and Networking Conference, 2007. WCNC 2007, IEEE*. Tree-based routing protocol for cognitive wireless access networks, (2007), pp. 4204–4208. doi:10.1109/WCNC.2007.768
14. X Wang, TT Kwon, Y Choi, in *Proceedings of the 2009 ACM Workshop on Cognitive Radio Networks*. CoRoNet '09. A multipath routing and spectrum access (mrsa) framework for cognitive radio systems in multi-radio mesh networks (ACM, New York, NY, USA, 2009), pp. 55–60. doi:10.1145/1614235.1614249. http://doi.acm.org/10.1145/1614235.1614249
15. S-C Lin, K-C Chen, in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. Spectrum aware opportunistic routing in cognitive radio networks, (2010), pp. 1–6. doi:10.1109/GLOCOM.2010.5683924
16. W Li, X Cheng, T Jing, X Xing, in *INFOCOM, 2013 Proceedings IEEE*. Cooperative multi-hop relaying via network formation games in cognitive radio networks, (2013), pp. 971–979. doi:10.1109/INFCOM.2013.6566886
17. J Huang, S Wang, X Cheng, M Liu, Z Li, B Chen, Mobility-assisted routing in intermittently connected mobile cognitive radio networks. IEEE Trans. Parallel Distributed Syst. **25**(11), 2956–2968 (2014). doi: 10.1109/TPDS.2013.291
18. S Biswas, R Morris, in *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. SIGCOMM '05. Exor: opportunistic multi-hop routing for wireless networks (ACM New York, NY, USA, 2005), pp. 133–144. doi:10.1145/1080091.1080108. http://doi.acm.org/10.1145/1080091.1080108
19. S Chachulski, M Jennings, S Katti, D Katabi, in *Proceedings of the 2007 Conference on, Applications, Technologies, Architectures, and Protocols for Computer Communications*. SIGCOMM '07. Trading structure for randomness in wireless opportunistic routing (ACM, New York, NY, USA,

2007), pp. 169–180. doi:10.1145/1282380.1282400. http://doi.acm.org/10.1145/1282380.1282400

20. A Jindal, K Psounis, Contention-aware performance analysis of mobility-assisted routing. IEEE Trans. Mobile Comput. **8**(2), 145–161 (2009). doi:10.1109/TMC.2008.98
21. S Wang, X Wang, X Cheng, J Huang, R Bie, in *IEEE ICDCS*. The tempo-spatial information dissemination properties of mobile opportunistic networks with levy mobility (IEEE, New York, 2014)
22. S Wang, M Liu, X Cheng, Z Li, J Huang, B Chen, Opportunistic routing in intermittently connected mobile p2p networks. IEEE J. Selected Areas Commun. (JSAC). **31**(9), 369–378 (2013)
23. Y Duan, G Liu, Z Cai, in *Wireless Algorithms, Systems, and, Applications*. Opportunistic channel-hopping based effective rendezvous establishment in cognitive radio networks (Springer, New York, 2012), pp. 324–336
24. X Lin, N. B Shroff, in *INFOCOM 2005. 24th Annual, Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3. The impact of imperfect scheduling on cross-layer rate control in wireless networks, (2005), pp. 1804–18143. doi:10.1109/INFCOM.2005.1498460
25. A Dimakis, J Walrand, Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits. Adv. Appl. Probability. **38**(2), 505–521 (2006)
26. L Tassiulas, Adaptive back-pressure congestion control based on local information. Automatic Control IEEE Trans. **40**(2), 236–250 (1995). doi:10.1109/9.341781
27. MJ Neely, E Modiano, CE Rohrs, Dynamic power allocation and routing for time-varying wireless networks. Selected Areas Commun. IEEE J. **23**(1), 89–103 (2005). doi:10.1109/JSAC.2004.837349(410)23
28. X Wu, R Srikant, JR Perkins, Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. Mobile Comput. IEEE Trans. **6**(6), 595–605 (2007). doi:10.1109/TMC.2007.1061
29. L Jiang, J Walrand, A distributed csma algorithm for throughput and utility maximization in wireless networks. Netw. IEEE/ACM Trans. **18**(3), 960–972 (2010). doi:10.1109/TNET.2009.2035046
30. L Georgiadis, MJ Neely, L Tassiulas, Resource allocation and cross-layer control in wireless networks. Found. Trends Netw. **1**(1), 1–144 (2006). doi:10.1561/1300000001
31. MJ Neely, E Modiano, CE Rohrs, in *IEEE Journal on Selected Areas in Communications*. Dynamic power allocation and routing for time varying wireless networks (IEEE, New York, 2003), pp. 89–103
32. MJ Neely, E Modiano, C-P Li, in *Proceedings of IEEE INFOCOM*. Fairness and optimal stochastic control for heterogeneous networks (New York, 2005), pp. 1723–1734
33. L Huang, S Moeller, MJ Neely, B Krishnamachari, in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt) 2011 International Symposium On*. Lifo-backpressure achieves near optimal utility-delay tradeoff, (2011), pp. 70–77. doi:10.1109/WIOPT.2011.5930067
34. Y Le, L Ma, W Cheng, X Cheng, B Chen, in *INFOCOM, 2012 Proceedings IEEE*. Maximizing throughput when achieving time fairness in multi-rate wireless lans, (2012), pp. 2911–2915. doi:10.1109/INFCOM.2012.6195728
35. M Heusse, F Rousseau, R Guillier, A Duda, in *SIGCOMM '05: Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. Idle Sense: an optimal access method for high throughput and fairness in rate diverse wireless lans (ACM, New York, NY, USA, 2005), pp. 121–132
36. S Biswas, R Morris, Opportunistic routing in multi-hop wireless networks. SIGCOMM Comput. Commun. Rev. **34**(1), 69–74 (2004). doi:10.1145/972374.972387
37. OMNeT++ Home Page. http://www.omnetpp.org