

Anomalies in Optimal Rate-control and Scheduling Protocols for Cognitive Radio Networks

Vinay Kolar

Department of Computer Science
Carnegie Mellon University, Qatar.
vkolar@cmu.edu

Vikram Munishwar and Nael B. Abu-Ghazaleh

Department of Computer Science,
State University of New York, Binghamton.
{vmunish1,nael}@cs.binghamton.edu

Abstract—Protocols in Cognitive Radio Networks (CRNs) tune the radio and network parameters to utilize the unused frequency spectrum, and thus improve the application performance. Generalized Network Utility Maximization (GNUM) approach has been used to derive optimal algorithms in CRN from a theoretical formulation of the problem. However, there are several practical challenges for realizing these algorithms in a networking system. In this paper, we consider a joint source-rate, routing and scheduling GNUM problem in CRNs. We discuss three issues of protocols that are derived directly from the optimal algorithms: spurious pressure points, bursty scheduling and the need for link pruning. We show that the above practical problems result in serious system implications such as large message passing overheads, packet delays and buffer requirements.

I. INTRODUCTION

Frequency spectrum is scarcely used in several licensed spectrum bands such as TV white-spaces [1], while the license-free band, which are used by WiFi networks, are heavily congested. Cognitive Radio Networks (CRNs) take advantage of the unused spectrum in licensed bands; they dynamically use the spectrum bands which are not used by licensed primary users. CRNs are equipped with nodes with flexible radios that adjust low-level radio and PHY-layer parameters to operate over a wide spectrum band. Thus, CRNs provide a promising solution to alleviate the capacity shortage issues in wireless networks, such as WiFi, that operate in license-free bands.

Large spectrum does not always translate into large capacity in CRNs. The primary user activity may forbid the use of a spectrum band or policy restrictions may block the usage of some parts of the spectrum. In addition, the growing capacity demands by wireless applications, which has been a prominent trend in recent years, may exceed the CRN capacity limits. The network is expected to respect the constraints of policy and primary users, and efficiently use the available networking resources to provide better performance to the applications.

Optimization of application performance is a hard problem in CRNs since it depends upon several radio and networking parameters [2]. Existing heuristic protocols search for efficient and low-complexity solutions. However, it is challenging to systematically and analytically quantify the best solution. Such systematic solution is necessary not only to predict the theoretically optimal solution, but also towards designing practical optimal protocols that ensure near-optimal solutions.

Generalized Network Utility Maximization (GNUM) has been extensively used to optimally adjust the network parameters, and derive optimal algorithms [3]. GNUM states the application optimization as a convex optimization problem [4], and, based on the structure of the problem, derives algorithms that are dynamic, distributed and converge to optimal solution. Joint source-rate control, routing and scheduling is one the key problems in GNUM which has been widely studied in traditional wireless networks [5], [6] and CRNs [7], [8].

Translating the optimal algorithms derived from GNUM to protocols in CRN system poses practical challenges. The theoretical problems assume ideal conditions, such as low message passing overheads, for the simplicity of formulation. Realizing optimal algorithms into protocols require evaluating the assumptions under a system perspective such that the protocol has near-optimal performance.

In this paper, we analyze the practical anomalies in a joint rate-control, routing and scheduling GNUM problem in CRNs. We describe the general network model in Section II, which has been extensively studied in the literature [7], [6]. The main contribution of the paper, which is described in Section III, is identifying key practical considerations for the design of near-optimal protocols. First, we show that optimal algorithms lead to accumulation of packets at inactive nodes; we call this phenomenon as *spurious pressure points*. Second, we illustrate the effect of *bursty scheduling*. Finally, we motivate the need for *link pruning*, where unnecessary links are ignored by the scheduler, to design fast schedulers.

Through numerical examples in different scenarios, we show that the anomalies of the optimal algorithms lead to practical networking issues such as message passing overheads, large packet delays, retransmissions and large buffer requirements. Finally, we conclude and discuss the future work in Section IV.

II. GENERAL NETWORK MODEL

In this section, we formulate the joint rate-control, routing and scheduling problem to a network with frequency-agile radios. We extend the general single channel model proposed by Chen *et. al.* [6], and use the similar notations where ever applicable.

We represent the network as a set of nodes (\mathcal{N}) and links (\mathcal{L}). A link between nodes a and b is represented by (a, b)

or by an index i . Each node is equipped with radios that can operate on a wide frequency band. The band is divided into set of discrete channels (\mathcal{C}). The radio can transmit or receive on a subset of these discrete channels. Such a radio model is reasonable for OFDMA based systems, where the frequency band is divided into a number of sub-carriers, and a radio can transmit or receive on a subset of sub-carriers.

In networks operating on a wide-frequency band, the quality of links vary depending upon the channel frequency. OFDMA based systems allow to modulate each sub-carrier separately (say, based on link quality). The time to transmit unit data on a link i on channel c ($T_{i,c}$) is hence a function of the chosen modulation.

The set of end-connections in a network is represented by \mathcal{K} . The source, destination and the rate of transmission for a connection k is denoted by $\text{src}(k)$, $\text{dest}(k)$ and r_k , respectively. Each connection k has an application utility that is expressed by a convex utility function $U(r_k)$. We use the utility function $U(r_k) = \log(r_k)$, which ensures proportional fairness to connections. Other convex utility functions can be used for ensuring different application utilities [9].

A link may carry traffic for various connections on different channels. The traffic flow allocated to link i (or (a,b)) on channel c for connection k is denoted by $f_{i,c}^k$ (or $f_{ab,c}^k$). The time required to transmit this flow over the link i on channel c is a function of the link capacity, and is denoted by $t_{i,c}^k = T_{i,c} f_{i,c}^k$. The vector of various variables (say, f , t and r) are represented by their bold letter counterparts (\mathbf{f} , \mathbf{t} and \mathbf{r}).

We now discuss the primal and dual optimization problem to optimize \mathbf{r} and \mathbf{f} for a given network topology. Finally, we briefly discuss the algorithm to solve the dual optimization problem with an example scenario.

A. Primal problem

The primal problem allocates flows on various links and channels such that: (1) the application utilities are maximized, (2) packets are routed from sources to the destinations, and (3) the schedule for link transmission is feasible. The overall problem is given by:

$$\text{Maximize } \sum_{\forall k \in \mathcal{K}} U(r_k), \quad (1)$$

such that

$$\mathbf{f} \geq 0, \quad (2)$$

$$x_a^k \leq \sum_{b:(a,b) \in \mathcal{L}} \sum_{c \in \mathcal{C}} f_{ab,c}^k - \sum_{b:(b,a) \in \mathcal{L}} \sum_{c \in \mathcal{C}} f_{ba,c}^k, \quad (3)$$

$$\forall k \in \mathcal{K}, \forall a \in \mathcal{N}, a \neq \text{dest}(k),$$

$$\mathbf{f} \in \Pi. \quad (4)$$

Equations 1 and 2 are obvious. Equation 3 represents the mass balance constraints, which is used to route packets from source to destination. Here, the source of a connection k injects a flow into the network, and is indicated by a positive demand variable $x_{\text{src}(k)}^k = r_k$. Similarly, the destination has negative demand $x_{\text{dest}(k)}^k = -r_k$. For other nodes a , $x_a^k = 0$.

Equation 4 describes the scheduling constraint that a schedule allocated on different link-channel combinations belongs to a feasible region Π . A flow allocation \mathbf{f} on different links and channels is feasible when certain properties are satisfied:

(1) *Primary Usage Constraint*: If a link i observes primary user activity on channel c for time-fraction $P_{i,c}$, then a link may schedule only for the remaining period (Equation 5).

(2) *Maximal Independent Set (MIS) constraint*: The flows on links on different channels have to be scheduled in cognitive wireless networks such that links do not interfere with each other. Interference in cognitive radio wireless network is represented by a *multi-channel conflict graph* [7], where each link-channel pair $(\langle i, c \rangle)$ is represented as a vertex of conflict graph and an edge between two vertices exist if they interfere. A Maximal Independent Set (MIS) of this graph is a set of link-channel pairs $\langle i, c \rangle$ that can be concurrently scheduled. The set of all MIS is represented by $\mathcal{M} = \{M_1, M_2, \dots, M_I\}$, where M_n is an MIS and I is the total number of MIS for the conflict graph.

The scheduler assigns a fraction of time s_n to MIS M_n , thus allowing the transmissions on link-channel pairs in M_n . Equation 6 specifies the lower bound for MIS time-share, and the maximum cumulative share of time for all MIS is 1 (Equation 7). The schedule is feasible if the overall time-share for a transmitting over a link-channel pair $\langle i, c \rangle$ cannot exceed the sum of the time-shares of MIS to which it belongs (Equation 8). Hence, a schedule is feasible, $\mathbf{f} \in \Pi$, if:

$$\sum_{k \in \mathcal{K}} t_{i,c}^k + P_{i,c} \leq 1, \quad \forall i \in \mathcal{L}, c \in \mathcal{C}, \quad (5)$$

$$s_n \geq 0, \quad n = 1..I, \quad (6)$$

$$\sum_{n=1..I} s_n \leq 1, \text{ and} \quad (7)$$

$$\sum_{k \in \mathcal{K}} t_{i,c}^k \leq \sum_{n: \langle i, c \rangle \in M_n} s_n, \quad \forall i \in \mathcal{L}, c \in \mathcal{C}. \quad (8)$$

Note that under realistic wireless interference models computation of MIS is NP-hard, and polynomial approximations exist to compute such sets [10]. The number of MIS increases rapidly as the number of edges or channels grow. Hence, as we discuss in Section III-D, realistic computation of MIS for a network is a demanding task for a scheduler.

B. Dual problem

We formulate the dual problem by Lagrangian dual decomposition using the well-known GNUM framework [6]. We relax the balance constraints in Equation 3. For each constraint in Equation 3, we introduce a Lagrangian multiplier q_a^k . Since a node maintains a queue for each connection, it can be shown that q_a^k maps to the congestion *price* of the queue for connection k on node a . Higher value of price q_a^k indicates that queue length is larger.

Through standard derivation approach, the primal problem (Equations 1–4) can be shown to be decomposed into two

sub-problems:

$$D_1(\mathbf{q}) = \max_{\mathbf{r} \geq 0} \sum_{k \in \mathcal{K}} U(r_k) - \sum_{k \in \mathcal{K}} r_k q_{\text{src}(k)}^k, \quad (9)$$

and

$$D_2(\mathbf{q}) = \max_{\mathbf{f} \geq 0} \sum_{(a,b) \in \mathcal{L}} \sum_{c \in \mathcal{C}} f_{ab,c}^k \max_{k \in \mathcal{K}} (q_a^k - q_b^k),$$

such that $\mathbf{f} \in \Pi$. (10)

$D_1(\mathbf{q})$ is a source-rate maximization problem, which can be solved in a distributed manner; each connection source node ($\text{src}(k)$) executes a local maximization problem:

$$\max_{r_k \geq 0} U(r_k) - r_k q_{\text{src}(k)}^k.$$

$D_2(\mathbf{q})$ is a joint routing and scheduling problem. The objective is a weighted maximization problem with each flow on a link (a, b) has a weight $w_{ab} = \max_{k \in \mathcal{K}} (q_a^k - q_b^k)$. Since the weight on a link w_{ab} is the maximum difference of prices on sender and receiver of a link, it can be viewed as a *maximum differential price* for a link.

C. Optimal protocols

Protocols to dynamically solve the optimal problems is derived based on the sub-problems $D_1(\mathbf{q})$ and $D_2(\mathbf{q})$. At each time-slot all the nodes co-operate and perform different tasks to solve the sub-problems. It can be shown that the following protocol executed at each time-slot (t) solves the joint rate-control, routing and scheduling problem:

Step 1: Each node a updates the congestion price $q_a^k(t)$ as follows:

$$q_a^k(t) = q_a^k(t-1) + \alpha \left\{ x_a^k - \left(\sum_{(a,b) \in \mathcal{L}, c \in \mathcal{C}} f_{ab,c}^k - \sum_{(b,a) \in \mathcal{L}, c \in \mathcal{C}} f_{ba,c}^k \right) \right\}^+,$$

where α is a non-negative integer step-size parameter. The value the expression $\{p\}^+ = p$ if p is non-negative. Otherwise it is 0. Hence, if the queue has more incoming packets (either from incoming links or from application) than outgoing packets, then the congestion price for the queue will increase. Step 2: Each source node of a connection executes a local source-rate maximization based on $D_1(\mathbf{q}(t))$. Here, the source alters the rate based on the observed queue price and application utility.

Step 3: Scheduling and routing are performed by solving $D_2(\mathbf{q}(t))$. Each link (a, b) calculates the maximum price differential $w_{ab}(t) = \max(q_a^k - q_b^k)$ for all connections k , and marks the connection ($k_{\max_i}(t)$) for which the maximum price-differential is observed. The sender and receiver of link also transmit primary usage map $P_{i,c}(t)$, which is obtained from the spectrum sensing components. The information $w_{ab}(t), P_{i,c}(t)$ is sent to the centralized scheduler. The scheduler computes the optimal flow $\mathbf{f}(t)$ by solving $D_2(\mathbf{q}(t))$, and the link allocates the flow to the queue for the connection indicated by $k_{\max_i}(t)$.

Scheduling in step 3 works on the basis of back-pressure. The scheduler uses the price-differentials at various queues to prioritize the packets: transmission from the longest queues are

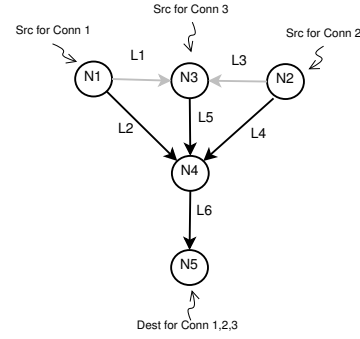


Fig. 1. 2-level tree scenario.

first scheduled before allocating schedules to other queues. By this procedure the algorithm schedules packets to neighboring nodes such that almost equal queue lengths (back-pressures) are observed at neighboring queues.

While we describe dynamic protocols using a centralized scheduler, the framework can be readily extended to distributed protocols similar to [6]. The only additional information exchanged between the nodes is the spectrum occupancy by primary users $P_{i,c}(t)$ for each link.

III. PRACTICAL CONSIDERATIONS IN PROTOCOL DESIGN

In this section, we first illustrate the optimal protocol described in Section II-C by using numerical example. Then, we identify the important limitations of optimal protocols, and illustrate them with simple examples.

A. Example scenario

Consider a 2-level tree scenario with three connections as shown in Figure 1. Each node operates on two channels, and there is no interference from primary transmitters on either channels. Each channel is assumed to have unit capacity, thus each link has a capacity of 2 units.

We run the optimal protocols in Section II-C for 500 time-slots with no primary traffic ($P_{i,c} = 0$), $\alpha = 0.1$ and $T_{i,c} = 1$. The source-rate of the connections smoothly converge close to the optimal value 0.33 as the time increases (similar to Figure 2(c), but without the intermediate turbulence observed from time-slots 300 to 500). The average of the flows allocated to links on different channels is shown in Figure 2(a).

We re-run the scenario with primary activity as marked in Figure 2(c). It can be seen that primary user activity creates small turbulence in the instantaneous rate. However, the rates dynamically adapt and converge over time.

Figures 2(a) shows the average flows in this scenario. Even though $L2$ has one of the channel unusable for the whole period, there is no drop in any source-rates. The intelligent scheduler subdues the effect of primary activity observed at $L2$. It allocates $L2$ a flow of 0.33 on unused channel, instead of allocating flow of 0.17 on two channels (as seen by comparing Figures 2(a) and 2(b)). Hence, if feasible, the scheduler re-arranges link schedules such that source-rates are not reduced.

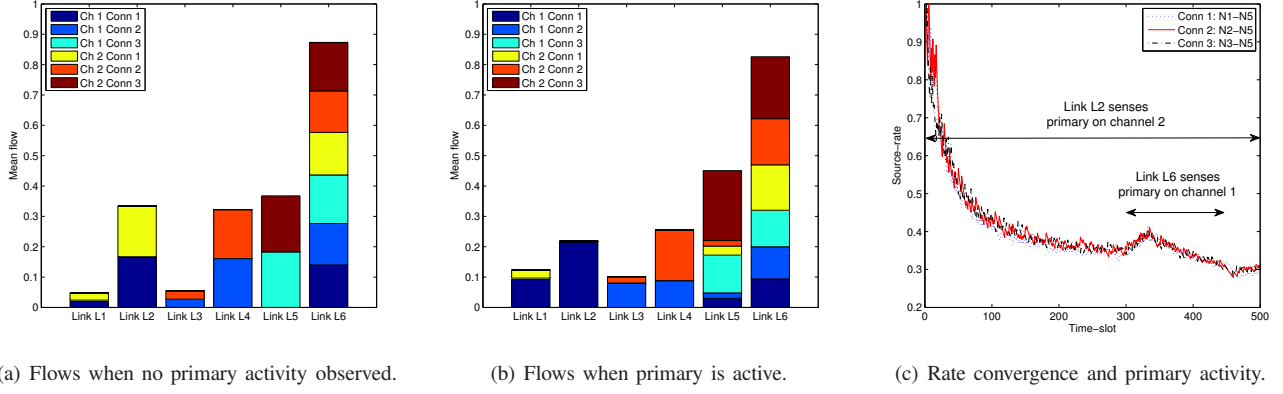


Fig. 2. Rate convergence, mean flows and the effect of primary user activity in a 2-level tree scenario.

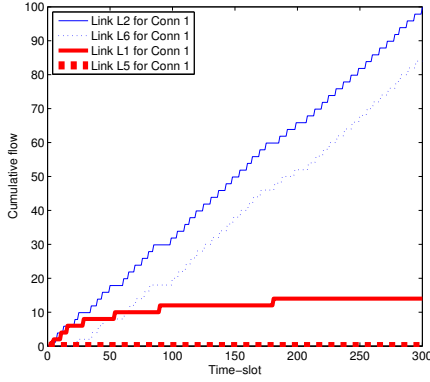


Fig. 3. Spurious pressure point: Flow for connection 1 accumulate at node $N3$ (on link $L1$), which is inactive in routing.

B. Spurious pressure points

Recall that the building of back-pressure enables link scheduling. The bootstrapping process required for back-pressure algorithm is building enough pressure at all neighboring nodes – even on the nodes inactive in routing. For example, node $N1$ can send the packets for connection 1 to $N4$ or $N3$. If there are already some packets at $N4$, the price-differential of link $L2$ becomes lower than that of $L1$. Hence, $N1$ pushes packets to $N3$ on link $L1$. However, $L5$ prefers transmitting packets of connection 2 for providing proportional fairness to connections (Equation 9). Hence, packets of connection 1 accumulate at $N3$ for a long time, as shown in Figures 3 and 2(a). Spurious pressure results in several higher-layer issues such as out-of-order delivery, retransmissions of packets and waste of queue buffers.

C. Bursty schedules

Recall that the ideal scheduler chooses the packets from the queues that have maximum price differential. This process results in bursty schedules. For example, Figure 4 shows the schedules chosen for connection 1 on $L1$ and $L6$ of Figure 1.

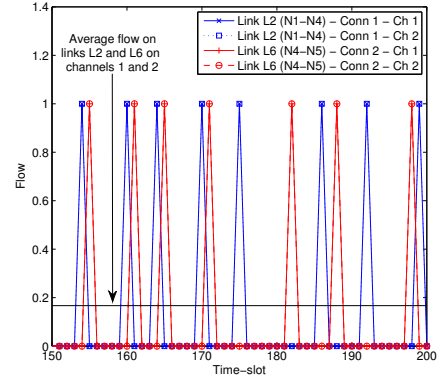


Fig. 4. Bursty schedules: For connection 1, links $L2$ and $L6$ are scheduled in bursts. The long-term average of the flow is constant.

If the scheduling and price updates happen at a fine time-granularity, then the inter-packet delay is reasonably small. Hence, the effect of bursty schedules can be ignored. However, the message passing overhead outruns the advantages of maximizing the source-rate if we disseminate the prices at a fine time-granularity. In reality, the price updates are chosen at, say, fraction of seconds to few seconds to reduce the dissemination overhead, and the time-slot to schedule the packet is chosen as few milliseconds.

Consider the back-of-the-envelope calculations for the scenario in Figure 1. Assume that the two channels have a capacity of 1 Mbps each. The price updates occur every second and packet transmission time-slot is 2 ms. Packet sizes are chosen to fit into the time-slot (500 bytes). We map the fraction of flow as solved by the optimal algorithms into packets, and record the end-to-end delays of packets. Figure 5 shows the large end-to-end delay (≈ 80 seconds) is observed, which has severe limitations in realistic networking systems: they harm performance of many applications, result in unfairness and require large buffer spaces at intermediate nodes.

Bursty schedules are eliminated by using heuristics (such

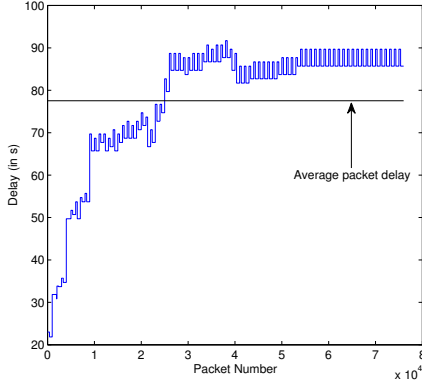


Fig. 5. Packet end-to-end delays: Prohibitively large delays are observed due to bursty schedules.

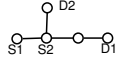
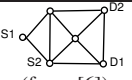
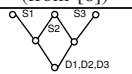
Scenario	Metric	Without Link Pruning	With Link Pruning
Figure 1	# Links, # MIS	12, 70	4, 10
	Primal Src-rates	0.33,0.33,0.33	0.33,0.33,0.33
	Dual Src-rates	0.35,0.35,0.36	0.36,0.35,0.34
	# Links, #MIS	8, 26	4, 3
	Primal Src-rates	0.5,1	0.5,1
	Dual Src-rates	0.53,0.87	0.53,0.87
 (from [6])	# Links, #MIS	20, 468	7, 27
	Primal Src-rates	0.5,1	0.5,1
	Dual Src-rates	0.55,0.91	0.53,0.86
	# Links, #MIS	12, 168	6, 17
	Primal Src-rates	0.67,0.67,0.67	0.67,0.67,0.67
	Dual Src-rates	0.72,0.62,0.66	0.70,0.67,0.62

TABLE I

EFFECT OF LINK PRUNING: LINK PRUNING REDUCES THE NUMBER OF EDGES AND MIS WITHOUT COMPROMISING THE PERFORMANCE.

as scheduling the long-term average, instead of instantaneous flow). Here, the convergence of source-rates to optimal values is not guaranteed. We wish to pursue heuristics to achieve near-optimal rates in the future work.

D. Link pruning

Recall that the construction of MIS, which is required for scheduling, is an NP-hard problem. Hence, it takes prohibitively long time to calculate the schedules as the edges increase. The effect is even more exacerbated in CRNs, where the number of channels are large. Hence, eliminating surplus edges that are not active in routing is beneficial in faster scheduler algorithms.

Fast schedulers can be realized in networking systems by decoupling routing from rate-control and scheduling problem. Most systems employ link-state or source-routing protocols, which already store the possible next-hops or entire routes. Using this information, link pruning eliminates the surplus edges that are inactive in routing. Link pruning also helps to reduce the effect of spurious pressure points, which is described in Section III-B, since the links to nodes that are inactive in routing have been pruned.

We implement link pruning by extended Dijkstra algorithm that computes all the minimum hop routes between given source-destination pair. We later prune all inactive edges. While we demonstrate the effect of link pruning using above algorithm, link pruning can be extended to more realistic routing protocols, such as link-quality based routing.

We simulate various scenarios as shown in Table I. The source and destination of a connection are marked as S_n and D_n . We compare the number of edges and maximum independent sets (MIS). We also compare the source-rates of optimal protocols with and without pruning. The results show that link pruning significantly reduces the number of edges and MIS with no loss in performance.

IV. CONCLUSIONS AND FUTURE WORK

The paper analyzes the issues in realizing optimal protocols for joint rate-control and scheduling in Cognitive Radio Networks. We identify three considerations: (1) spurious pressure points, which are built due to back-pressure algorithms, (2) bursty scheduling, due to maximum price-differential based scheduling, and (3) high-complexity schedulers due to large number of unnecessary links. The paper shows the above effects lead to large messaging overheads, packet delays and buffer requirements. In our future work, we plan to realize the near-optimal protocols in testbeds.

ACKNOWLEDGEMENTS

This research work was funded by Qatar National Research Fund (QNRF) under the National Priorities Research Program (NPRP) Grant No.:08-562-1-095.

REFERENCES

- [1] M. A. McHenry, "NSF Spectrum Occupancy Measurements Project Summary," Shared Spectrum Company, Technical Report, 2005.
- [2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127 – 2159, 2006.
- [3] M. Chiang, S. Low, A. Calderbank, and J. Doyle, "Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255 – 312, jan. 2007.
- [4] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.
- [5] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability," *The Journal of the Operational Research Society*, vol. 49, no. 3, pp. pp. 237–252, 1998.
- [6] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-Layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks," apr. 2006, pp. 1 – 13.
- [7] J. Tang, S. Misra, and G. Xue, "Joint spectrum allocation and scheduling for fair spectrum sharing in cognitive radio wireless networks," *Comput. Netw.*, vol. 52, no. 11, pp. 2148–2158, 2008.
- [8] Y. Hou, Y. Shi, and H. Sherali, "Spectrum Sharing for Multi-Hop Networking with Cognitive Radios," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 1, pp. 146 – 155, jan. 2008.
- [9] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, 2000.
- [10] H. Li, Y. Cheng, C. Zhou, and P. Wan, "Multi-dimensional Conflict Graph Based Computing for Optimal Capacity in MR-MC Wireless Networks," jun. 2010, pp. 774 – 783.