Optimizing Flow Control in Multi-interface Wireless Cognitive Radio Networks

Alireza Louni
IEEE Student Member
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ, USA.
Email: alouni@stevens.edu

K. P. Subbalakshmi
IEEE Senior Member
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ, USA.
Email: ksubbala@stevens.edu

***Abstract*—This paper considers multiple self-interested cognitive radios and multiple self-interested servers with multiple interfaces and solve the optimal flow management problem arising in this network. We formulate a novel flow management problem in which cognitive nodes compete to minimize the total delay over all interfaces while at the same time the servers compete to maximize their individual profit. We model this problem as a leader-follower game and propose a dynamic linear pricing scheme designed to achieve an optimal flow allocation. We propose an iterative algorithm to solve the game and analyze the criteria for convergence to the unique Nash Equilibrium. The messaging required to implement this algorithm is minimal thereby making it suitable for distributed implementations. Numerical simulations demonstrate significant improvement in terms of average total delay for cognitive nodes in comparison with alternative algorithms. Simulation results show that when quality of service in terms of average total delay is fixed, our algorithm improves the capacity of the network by 40% for the maximum allowable throughput demand.**

I. Introduction

Several innovative solutions have been formulated under the general umbrella of cognitive networking to address the problem of bandwidth shortage in wireless communications. One such method is to offload low quality of service (QoS) data traffic to WiFi while keeping the rest of the traffic on the long term evolution (LTE) network [1]. Another solution is to cognitively and dynamically handover communication between different interfaces based on interface/link quality [2]. The third Generation Partnership Project (3GPP) community has proposed a solution for a seamless WiFi offloading, like IP flow mobility and seamless offload (IFOM) [3]. Aggregated data is sent to a proxy node/server (e.g. “Amazon cloud”) [1] by using different IP addresses depending on the number of interfaces. At the proxy node, these IP addresses are mapped to a unique IP address and then sent to the final destination (e.g. “www.google.com”) [2].

Since interfaces are shared and customers are self-interested, behavior of each customer affects the rest of the network. Selfishness and a lack of co-operation could lead to non-optimal resource sharing. This effect can be mitigated if the servers offer the customers some incentives to share the resources more co-operatively. In a distributed environment, pricing (set by the network) can be used as an approach to achieve this goal. This type of selfish, uncooperative situation with a distinct leader leads to a hierarchical decision problem.

[1] Proxy node is employed for reordering streams from different interfaces and making transparency between the server and the multi-interface structure.
[2] In this model, proxy and server could be implemented at same node.

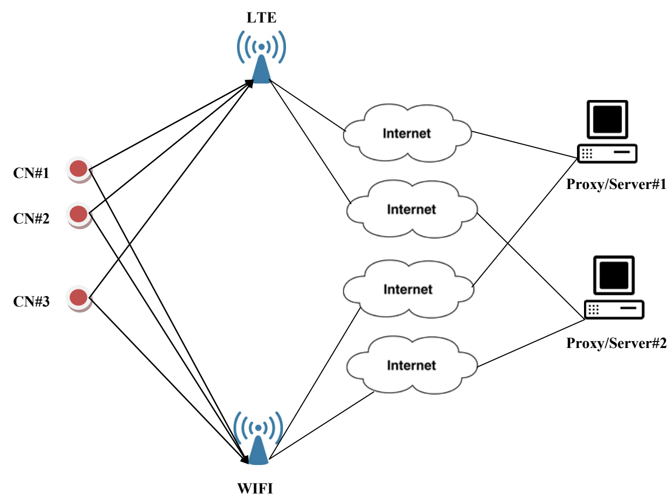

Fig. 1. Example for the system model: two servers, two interfaces to each provider (through LTE and WiFi), and three cognitive nodes. Each flow has four possible routes to its destination.

In this paper, a network of multiple cognitive nodes (CNs), and multiple servers is considered in which the throughput demand can be split through different interfaces of different servers. Cognitive nodes are decision makers that are capable of identifying individual data flows and making dynamic flow allocation decisions. Real world examples of such cognitive node would include smart phones that are capable of supporting more than one unique IP address and thereby able to differentiate between flows. Unlike previous works, we consider two levels of competition: the servers that compete to provide more throughput and the CNs that compete to optimize their performances (Fig. 1). We model this problem as a Stackelberg game and analyze the optimal routing solution.

We also design a novel, linear pricing scheme to achieve global optimality (servers and CNs) in the routing problem. We then analyze the convergence of the proposed algorithm and compare it to other methods. The amount of signaling necessary for the implementation of this algorithm is minimal, which allows a distributed implementation.

The flow control problem in a network with single network manager and multiple followers was studied in [4] and [5] and the problem of down-link flow allocation between multiple self-interested service providers was analyzed in [6]. To the best of our knowledge, this paper is the first work that considers flow management in the context of multi-interface wireless cognitive radio networks with multiple customers and multiple servers.

This paper is organized as follows. In Section II, system model and game problem formulation is defined. Linear pricing is designed to reach optimal flow allocation in Section III. Multi-stage algorithm is proposed and convergence criteria are analyzed as well in Section IV. Finally, numerical results and conclusion are provided in Section V and VI respectively.

II. System Model and Game Problem Formulation

A. System Model

We consider a set $\mathbb{M} = \{1, 2, ..., M\}$ of servers and a set $\mathbb{N} = \{1, 2, ..., N\}$ of CNs. Multi-interface structure is assumed to facilitate communications and split the throughput demand. Each server can send data via a set $\mathbb{L} = \{1, 2, ..., L\}$ of interconnecting interfaces. Let $C_{l,m}$ be the capacity of interface l of the server m. The throughput demand for each CN is Poisson distributed with average rate λ_i. $f^i_{l,m}$ denotes the fraction of the demand λ_i that CN i sends over the interface l of the server m. Hence, the flow control matrix for the i^{th} CN is denoted by an $L \times M$ matrix given by $F_i = (f^i_{l,m})$ and limited by action space

$$A_i = \{f^i_{l,m} \in \mathbb{R} : 0 \le f^i_{l,m} \le 1, \sum_m \sum_l f^i_{l,m} = 1, \sum_j \lambda_j f^j_{l,m} < C_{l,m}\}^3. \quad (1)$$

In this model, the CN can acquire service from several servers via multiple interfaces concurrently. In practice, the system architecture may include either a single proxy server or multiple proxy servers. We assume a multiple proxy server setting (Fig. 1) because it is the more general case and avoids a single point of failure.

Each interface is modeled as an M/M/1 queuing system. Hence, the resulting delay per unit of flow for the l^{th} interface of the m^{th} server) is given by [7]

$$\frac{1}{C_{l,m} - \sum_j \lambda_j f^j_{l,m}}. \quad (2)$$

And the total delay is

$$\frac{\lambda_i f^i_{l,m}}{C_{l,m} - \sum_j \lambda_j f^j_{l,m}}. \quad (3)$$

Each CN manages its demand non-cooperatively to minimize its total delay. This value, however, depends on actions taken by the rest of network. Therefore, the total delay can be considered as the utility function with negative sign to model selfish behavior. The utility of the i^{th} CN is given by

$$u_i(F_i, F_{-i}) = -\sum_m \sum_l \frac{\lambda_i f^i_{l,m}}{C_{l,m} - \sum_j \lambda_j f^j_{l,m}} - \sum_m \sum_l \lambda_i f^i_{l,m} P_{l,m} \quad (4)$$

where $f^i_{l,m}$ belongs to the set A_i defined in Eq. (1) and $P_{l,m}$ is the unit price charged by the server m using the profit function, $\gamma_m = \sum_l \sum_j \lambda_j f^j_{l,m} P_{l,m}$.

B. Game Problem Formulation

As mentioned above, CNs are selfish decision makers who compete to optimize their performances (see Eq.(4)) in a non-cooperative way. We model this problem as a Stackelberg game and study the system stability, optimality and uniqueness of the Nash Equilibrium (if it exists). The proposed game is represented mathematically by

$$G = < \mathbb{N}, \{A_i\}_{i \in N}, \{u_i\}_{i \in N} > \quad (5)$$

where $\mathbb{N}$ is the set of players (CNs), $A = A_1 \times A_2 \times ... \times A_N$ is the action space of players reflecting their flow allocation strategy and $u_i : A \to \mathbb{R}$ is the utility function defined in (4). The competition consists of two stages, first, prices ($P_{l,m}$) are announced by servers (leaders) as a $L \times M$ matrix, maximizing their utilities (γ_m). At the second stage, the flow allocation strategy is selected as a $L \times M$ flow control matrix, $F^{(i)}$, that maximizes the utility in (4) constrained by A_i.

The optimization problem at each stage for the i^{th} CN is given as follows

$$\max_{F_i} \{G_i(-\sum_m \sum_l \frac{\lambda_i f^i_{l,m}}{C_{l,m} - \sum_k \lambda_k f^k_{l,m}}) - \sum_m \sum_l \lambda_i f^i_{l,m} P_{l,m}\}$$
$$s.t. \quad f^i_{l,m} \in A_i \quad (6)$$
$$\forall i \in \mathbb{N}, \forall m \in \mathbb{M}, \forall l \in \mathbb{L}$$

where $G_i(.)$ is a concave and increasing function. Note that the utility is a monotonically increasing function with respect to $f^i_{l,m}$ when the other CNs' actions are fixed. Consequently, even if such a game converges, flow allocation leads to an inefficient NE in terms of whole network performance. In the next section, dynamic pricing is proposed to overcome this problem.

III. Pareto-optimal Pricing Design and Resource Allocation

A. Pareto-optimal Pricing Design

Although cost of service, $P_{l,m}$, controls selfishness slightly (since it is a pricing function);however, the NE of the system is still far away from Pareto optimality. The NE is defined as a point that no user has the incentive to change unilaterally[8]. The NE is Pareto optimal if by deviating from that NE, at least one CN loses its utility. To reach Pareto optimality, a linear pricing function, $t^i_{l,m}$, for the i^{th} CN is designed as follows

$$G_i(-\sum_m\sum_l \frac{\lambda_i f^i_{l,m}}{C_{l,m}-\sum_k \lambda_k f^k_{l,m}}) - \sum_m\sum_l \lambda_i f^i_{l,m} P_{l,m} - \sum_m\sum_l \lambda_i f^i_{l,m} t^i_{l,m}. \quad (7)$$

Proposition 1: Pricing factor to reach Pareto optimality is

$$t^i_{l,m} = \frac{N-1}{(C_{l,m}-\sum_{k\neq i} \lambda_k f^k_{l,m})^2} - P_{l,m}. \quad (8)$$

Proof: One mathematical way to measure how far our resource allocation is from Pareto optimality is by comparing it with the solution of the network utility maximization problem (NUM)[9]. Moreover, at NE, each user's strategy is obtained by solving Karush Kuhn Tucker (KKT) equations [10] given in (6) while other user' strategies are given. $t^i_{l,m}$ can be determined by equating KKT equations for the game (G) and the NUM. We calculate the Lagrangian function to find KKT equations as

$$\begin{aligned} &G_i(-\sum_m\sum_l \frac{\lambda_i f^i_{l,m}}{C_{l,m}-\sum_j \lambda_j f^j_{l,m}}) - \sum_m\sum_l \lambda_i f^i_{l,m}(P_{l,m}+t^i_{l,m}) \\ &+\sum_m\sum_l L^1_{l,m,i} f^i_{l,m} - L^2_i(\sum_m\sum_l f^i_{l,m}-1) \\ &-\sum_m\sum_l L^3_{l,m,i}(f^i_{l,m}-1) \end{aligned} \quad (9)$$

where $L^1_{l,m,i}$, L^2_i, $L^3_{l,m,i}$ are the Lagrangian multipliers to satisfy constraints in action space A_i. The KKT equations are obtained by differentiating the Lagrangian function with respect to $f^i_{l,m}$:

$$\begin{aligned} &\frac{\partial G_i(\omega_i)}{\partial \omega_i}\frac{\partial \omega_i}{\partial f^i_{l,m}} - P_{l,m}\lambda_i - t^i_{l,m}\lambda_i \\ &+ L^1_{l,m,i} - L^2_i - L^3_{l,m,i} = 0 \\ &f^i_{l,m} \geq 0 \\ &L^1_{l,m,i} f^i_{l,m} = 0 \\ &f^i_{l,m} \leq 1 \\ &L^3_{l,m,i}(1-f^i_{l,m}) = 0 \\ &L^1_{l,m,i}, L^2_i, L^3_{l,m,i} \geq 0 \end{aligned} \quad (10)$$

where $w_i \triangleq -\sum_m\sum_l \frac{\lambda_i f^i_{l,m}}{C_{l,m}-\sum_k \lambda_k f^k_{l,m}}$.

The NUM problem is provided by summation of servers' profit and the CN's utility

$$\sum_{i=1}^N G_i(-\sum_m\sum_l \frac{\lambda_i f^i_{l,m}}{C_{l,m}-\sum_k \lambda_k f^k_{l,m}}) - \sum_{i=1}^N\sum_m\sum_l \lambda_i f^i_{l,m} P_{l,m} + \sum_{i=1}^N\sum_m\sum_l \lambda_i f^i_{l,m} P_{l,m}. \quad (11)$$

The second and third terms in (11) cancel each other out. So, the NUM utility function is $\sum_{j=1}^N G_j(w_j)$[4] where $w_j \triangleq -\sum_m\sum_l \frac{\lambda_j f^j_{l,m}}{C_{l,m}-\sum_k \lambda_k f^k_{l,m}}$. Similarly, the Lagrangian function for the NUM problem is

$$\begin{aligned} L = &\sum_{i=1}^N G_i(-\sum_m\sum_l \frac{\lambda_i f^i_{l,m}}{C_{l,m}-\sum_j \lambda_j f^j_{l,m}}) \\ &+\sum_{i=1}^N\sum_m\sum_l L^1_{l,m,i} f^i_{l,m} - \sum_{i=1}^N L^2_i(\sum_m\sum_l f^i_{l,m}-1) \\ &-\sum_{i=1}^N\sum_m\sum_l L^3_{l,m,i}(f^i_{l,m}-1). \end{aligned} \quad (12)$$

By differentiating of the Lagrangian function with respect to $f^i_{l,m}$, KKT equations are given by

$$\begin{aligned} &\frac{\partial G_i(w_i)}{\partial w_i}\frac{\partial w_i}{\partial f^i_{l,m}} + \sum_{j\neq i}\frac{\partial G_j(w_j)}{\partial w_j}\frac{\partial w_j}{\partial f^i_{l,m}} + L^1_{l,m,i} - L^2_i - L^3_{l,m,i} = 0 \\ &f^i_{l,m} \geq 0 \\ &L^1_{l,m,i} f^i_{l,m} = 0 \\ &f^i_{l,m} \leq 1 \\ &L^3_{l,m,i}(1-f^i_{l,m}) = 0 \\ &L^1_{l,m,i}, L^2_i, L^3_{l,m,i} \geq 0. \end{aligned} \quad (13)$$

By equating the two sets of KKT equations (10) and (13), $t^i_{l,m}$ can be obtained as follows

$$-\lambda_i(P_{l,m}+t^i_{l,m}) = \sum_{j\neq i}\frac{\partial G_j(w_j)}{\partial w_j}\frac{\partial w_j}{\partial f^i_{l,m}}. \quad (14)$$

W.l.o.g assuming $G(x) = x$ and substituting w_j in (14),

$$t^i_{l,m} = \frac{N-1}{(C_{l,m}-\sum_{k\neq i} \lambda_k f^k_{l,m})^2} - P_{l,m}. \quad (15)$$

In order to derive $t^i_{l,m}$, we need to know $(\frac{1}{C_{l,m}-\sum_k \lambda_k f^k_{l,m}})$ and the cost of service ($P_{l,m}$). This information can be obtained by servers and exchanged over a common channel without incurring too much communication overhead. Finally, with comparing our problem with the NUM problem and using low amount of information, the utility defined in Eq. (7) leads us to Pareto optimality.

[4]Priority based service also can be applied to our model by using weighted utility.

B. Resource Allocation

When other CNs' strategies are fixed, the optimum strategy (or the best response: $BR(F_{-i})$) of the i^{th} CN, $(f_{l,m}^{i})^{*}$, is given by

$$\left[\frac{1}{\lambda_i}(C_{l,m}-\sum_{k\neq i}\lambda_k f_{l,m}^{k}-\sqrt{\frac{\lambda_i\sum\limits_{k\neq i}\lambda_k f_{l,m}^{k}-\lambda_i C_{l,m}}{L_i^2+\lambda_i P_{l,m}+\lambda_i t_{l,m}^{i}}})\right]_0^1 \tag{16}$$

where $[.]_0^1$ implies that the value is bounded between 0 and 1. Note that the utility function in (6) is monotonically increasing and concave with respect to $f_{l,m}^{i}$. Therefore, KKT equations in our game (G) provide the optimum strategy. Again, the required information is $C_{l,m}-\sum\limits_{k\neq i}\lambda_k f_{l,m}^{k}$ which can be found as $C_{l,m}-\sum\limits_{k}\lambda_k f_{l,m}^{k}+\lambda_i f_{l,m}^{i}$ [5]. On the whole, amount of required information is not large and can be exchanged over a common channel.

The Lagrangian multiplier, L_i^2, can be updated in a distributed way by the dual sub-gradient method as follows

$$L_i^2(n_t)=\max\ \{\ 0, L_i^2(n_t-1)+\ \alpha_i(n_t)(\sum_{m}\sum_{l}f_{l,m}^{i}-1)\} \tag{17}$$

where $\alpha_i(n_t)=\frac{k}{\sqrt{n_t}}, k>0$.

One of the boundary condition is obtained for the case when the flow is routed through more than one interface, $0<f_{l,m}^{i}<1$. This implies that $L_{l,m,i}^{1}=L_{l,m,i}^{3}=0$. Using this in the KKT equations, gives

$$\frac{\partial G_i(w_i)}{\partial w_i}d_{l,m}=P_{l,m}\lambda_i+L_i^2. \tag{18}$$

where $d_{l,m}\triangleq\frac{\lambda_i\sum\limits_{k\neq i}\lambda_k f_{l,m}^{k}-\lambda_i C_{l,m}}{(C_{l,m}-\sum\limits_{k}\lambda_k f_{l,m}^{k})^2}$. The second boundary condition is when all flows are allocated through one server, as follows:

$$\sum_{l}f_{l,m}^{i}=1. \tag{19}$$

Substituting (16) in (19), corresponding conditions can be found. The intersection of two sets of conditions in (18) and (19) provides a new set of conditions in which throughput demand is routed via multiple interfaces of one server.

IV. Algorithm Design and Convergence Analysis

A. Algorithm Design

In this section, an iterative algorithm is proposed to reach NE for the flow allocation game (G) in (5). The proposed algorithm (Table I) is composed of two stages: in the first stage, cost of service is determined by the servers. In the second stage CNs compete by using best response action in (16) while $P_{l,m}$ is fixed so that criterion in (20) is satisfied for all CNs. The convergence criterion is met when the deviation of the flow allocation matrix is less than ε (the same criterion is assumed for the pricing factor function update, $t_{l,m}^{i}$)

$$n_t<n_{\max}\ \text{ or }\ \frac{\left\|F_i^{n_t}-F_i^{n_t-1}\right\|}{\left\|F_i^{n_t-1}\right\|}\le\varepsilon \tag{20}$$

where $F_i^{n_t}$ is the flow allocation matrix at the $n_t{}^{\text{th}}$ iteration and ε is a small constant. $n_{\max}$is the maximum allowable iteration. Note that update period for the first stage must be greater than the convergence time of the second stage.

TABLE I
Flow Allocation Algorithm

1 : Initialize $P_{l,m}$for $l\in\mathbb{L}$ and $m\in\mathbb{M}$.
2 : Initialize $f_{l,m}^{i}$ for all $i\in\mathbb{N}$, $l\in\mathbb{L}$, and $m\in\mathbb{M}$.
3 : Set $n_t=0$.
4 : until convergence criterion (20) is satisfied, repeat step 5
5 : update $f_{l,m}^{i}$ using equation (16) for $\forall l\in\mathbb{L}$ and $\forall m\in M$
6 : $n_t=n_{t+1}$
7 : $P_{l,m}$ is updated by servers.

B. Convergence Analysis

Theorem 1: At least one NE exists for the proposed game G in (6).

Proof: For the proposed game G, the conditions 1) the action space A_i is linear, compact and convex, 2) the utility function u_i is continuous in its action space, 3) the utility function is concave in its action space, are met. These conditions only guarantee the existence of an NE [11]; the algorithm may converge to different NEs for different runs. We now analyze the uniqueness of the NE.

Since pricing factor $t_{l,m}^{i}$ is time variable, it cannot be decoupled with the best response (see Eq. (16)) in game (G), making convergence analysis much more challenging. Similar to the convergence analysis in the iterative water-filling problem [12], game (G) is broken into inner game and outer game. The inner game updates flow allocation process and the outer game to update pricing factor $t_{l,m}^{i}$. We find the sufficient conditions for convergence.

We rewrite the optimum value in (16) as follows

$$(f_{l,m}^{i})^{*}=\left[\frac{1}{\lambda_i}(C_{l,m}-\sum_{k\neq i}\lambda_k f_{l,m}^{k}-\frac{1}{r_{l,m}^{i}})\right]_0^1 \tag{21}$$

$$r_{l,m}^{i}=\left(\frac{\lambda_i\sum\limits_{k\neq i}\lambda_k f_{l,m}^{k}-\lambda_i C_{l,m}}{L_2^i+\lambda_i P_{l,m}+\lambda_i t_{l,m}^{i}}\right)^{-1/2}. \tag{22}$$

Hereafter, we work with function of $t_{l,m}^{i}$, $r_{l,m}^{i}$, to track convergence analysis in a closed form. $(f_{l,m}^{i})^{*}$ can be shown as projection of the value $(-C_{l,m}+\sum\limits_{k\neq i}\lambda_k f_{l,m}^{k})$ over A_i and $r_{l,m}^{i}$:

$$(f_{l,m}^{i})^{*}=BR(F_{-i})=\left[-C_{l,m}+\sum_{k\neq i}\lambda_k f_{l,m}^{k}\right]_{r_{l,m}^{i}}^{A_i} \tag{23}$$

[5] $(\frac{1}{C_{l,m}-\sum_{k}\lambda_k f_{l,m}^{k}})$is the line congestion and can be measured by server at the end of line easily.

therefore, the matrix formulation will be

$$F_i = T(F_{-i}) = \left[-\widetilde{C} + \sum_{k\neq i}\lambda_k F_k\right]_{r^i}^{A_i} \tag{24}$$

where $\widetilde{C}$ is the capacity matrix with $C_{l,m}$ presenting element l by m. For fixed value of $r^i_{l,m}$, we must show that $\left\|T(F_i^1) - T(F_i^2)\right\|_{2,r^i} \leq \delta \left\|F_i^1 - F_i^2\right\|_{2,r^i}$, where δ lies between zero and one. $\|.\|_{2,w}$ Frobenius norm [6].

$$\begin{aligned}
&\left\|T(F_i^1) - T(F_i^2)\right\|_{2,r^i} \\
&= \left\| [-\widetilde{C} + \sum_{k\neq i}\lambda_k F_k^{(1)}]_{r^i}^{A_i} - [-\widetilde{C} + \sum_{k\neq i}\lambda_k F_k^{(2)}]_{r^i}^{A_i} \right\|_{2,r^i} \\
&\leq \left\| \sum_{k\neq i}\lambda_k (F_k^{(1)} - F_k^{(2)}) \right\|_{2,r^i} \\
&= \sum_{k\neq i}\sqrt{\sum_l \sum_m r^i_{l,m}{\lambda_k}^2 ((F_k^{(1)})_{l,m} - (F_k^{(2)})_{l,m})^2} \\
&= \sum_{k\neq i}\sqrt{\sum_l \sum_m r^k_{l,m}{\lambda_k}^2 ((F_k^{(1)})_{l,m} - (F_k^{(2)})_{l,m})^2 (\sqrt{\frac{r^i_{l,m}}{r^k_{l,m}}})^2} \\
&\leq \sum_{k\neq i}\max_k\{\lambda_k\sqrt{\frac{r^i_{l,m}}{r^k_{l,m}}}\}\sqrt{\sum_l \sum_m r^k_{l,m} ((F_k^{(1)})_{l,m} - (F_k^{(2)})_{l,m})^2} \\
&= \sum_{k\neq i}\max_k\{\lambda_k\sqrt{\frac{r^i_{l,m}}{r^k_{l,m}}}\}\left\|(F_k^{(1)} - F_k^{(2)})\right\|_{2,r^i}.
\end{aligned} \tag{25}$$

So, the inner game converges (for a fixed value of $r^i_{l,m}$) if $\sum_{k\neq i}\max_k\{\lambda_k\sqrt{\frac{r^i_{l,m}}{r^k_{l,m}}}\} < 1$. $r^i_{l,m}$can be updated as follows

$$r^i_{l,m}(n_t) = \omega r^i_{l,m}(n_t - 1) + (1-\omega)\widetilde{r}^i_{l,m}(n_t)^7 \tag{26}$$

where $\widetilde{r}^i_{l,m}(n_t)$ can be obtained via (22). ω is the memory factor. A large value of ω shows faster convergence at the expense of less robustness to system change. As proved in [13], this algorithm converges for $0 < \omega < 1$ with probability close to one. In our simulations ω has been set to four heuristically. Memory based algorithm acts in a sense as controller of $r^i_{l,m}$ and helps game to converge. This NE is Pareto optimal and so servers are not interested in deviating from the above NE. Hence, we conclude that the conditions for convergence still hold even for multi-shot update of cost of service.

[6]For $A \in C^{m\times n}$, the Frobenius norm is defined by $\|A\|_{2,w} = (\sum_i \sum_j w_{i,j}\, |a_{i,j}|^2)^{\frac{1}{2}}$.

[7]$r^i_{l,m}(.)$ is employed to obtain the optimum strategy in (21).

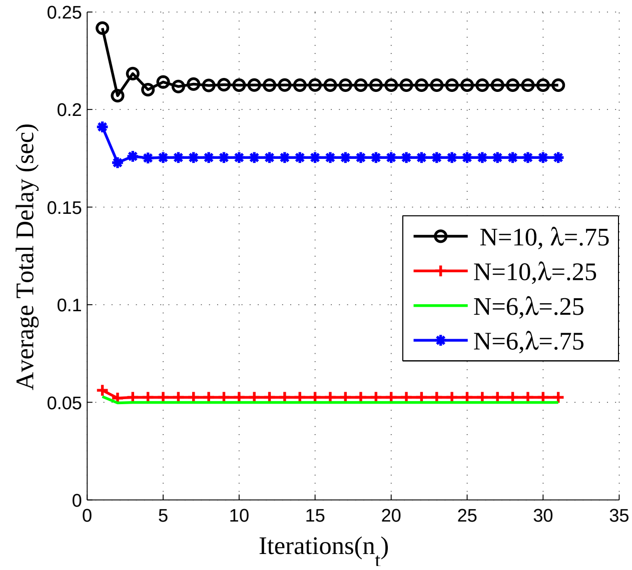

Fig. 2. Average total delay versus number of iterations. For larger values of the average demand or number of CNs, convergence requires more iterations.

V. Simulation and Numerical Results

In this section, we present the numerical results for the performance analysis of the proposed distributed algorithm. First, a network of two servers, four interfaces, and ten CNs similar to the system model in Fig. 1 is considered. Throughput demand for each CN is assumed to be Poisson distributed with average λ_i. Cost of service, $P_{l,m}$ is updated at one shot and fixed during convergence. The elements of the capacity matrix, $\widetilde{C}$, are set to be $C_{1,1} = 4$, $C_{1,2} = 7$, $C_{2,1} = 4.5$, $C_{2,2} = 6$.

The convergence rate of the proposed algorithm has been presented in Fig. 2. It illustrates average total delay over all interfaces as time goes on. The statistical average demand for all CNs is equal to λ. The figure shows that for larger values of the average demand or number of CNs, convergence requires more iterations. This is due to the fact that more number of CNs or average demand implies more competition. Moreover, the figure shows that at NE, light traffic inflicts lower amount of total delay to each CN.

Fig. 3 demonstrates the improvement in terms of total delay in our method to two methods: the best interface and uniform load. In the best interface method, all of the traffic is forwarded through the best interface (maximum value of $\frac{C_{l,m}}{P_{l,m}}$). The uniform load method refers to the case when the entire traffic is uniformly distributed over all available interfaces. Each curve in Fig. 3 represents the average of 1000 runs for $N = 10$ and 6 CNs. All graphs are shown inside valid regions defined by the capacity constraint and the convergence criteria. As shown in this figure, performance improvement over the best interface is remarkable except for small values of λ. This is expected, because at very low demands (λ) all traffic is forwarded through the best interface (since it can handle it). For the uniform load method, the load is split equally between the

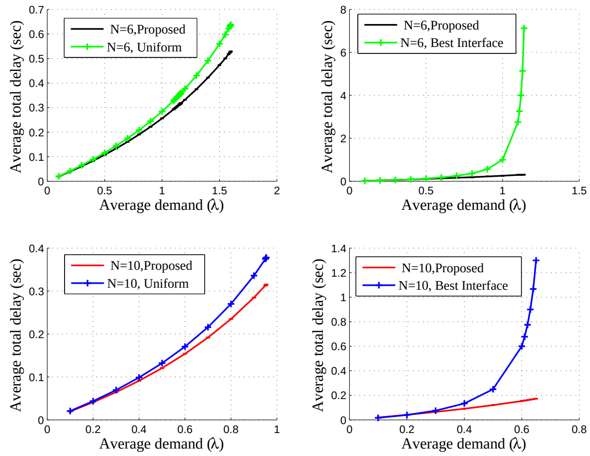

Fig. 3. Performance of the proposed algorithm vs. the best interface and the uniform methods : 20% improvement compared to the uniform method at high throughput demand for $\lambda = 1.6$ and $N = 6$.

four interfaces (25% per interface). For $\lambda = 1.6$ ($N = 6$), we get 20% improvement compared to the uniform load method in terms of the total delay. Let the round trip time (RTT) for the uniform method be RTT_1, and for our method, RTT_2,

$$RTT_1 = T_{u,1} + T_{d,1}$$

$$RTT_2 = T_{u,2} + T_{d,2}$$

where T_u and T_d are uplink and downlink delays respectively. Since the downlink is very small, $T_{d,1}$ and $T_{d,2}$ are nearly the same independent of the uplink traffic management method and much smaller than $T_{u,1}$ and $T_{u,2}$. Therefore,

$$\Delta RTT = RTT_2 - RTT_1 \cong T_{u,2} - T_{u,1} = \Delta T_u.$$

TCP utilization can be calculated as $u_{TCP} = \frac{W}{RTT}$ where W is the TCP window size [14]. So, Δu_{TCP} is given by

$$\Delta u_{TCP} = \frac{-W}{(RTT)^2} \times \Delta RTT = \frac{-W}{(RTT)^2} \times \Delta T_u.$$

Hence, improvement in sense of TCP would be 40%. If the quality of service is fixed by servers, this improvement can be translated into a 40% increase in the CNs that the network can support (network capacity).

VI. Conclusion

In this paper, a cognitive network consisting of a number of self-interested cognitive nodes(CNs), servers, and multiple interfaces is considered. All interfaces can be employed simultaneously. An optimization problem is defined to derive the best routing that achieves the lowest system delay.In this model, CNs compete selfishly to optimize their own utilities. A distributed algorithm is proposed to update the routing strategy. By applying game theoretical concepts, convergence and uniqueness of the Nash Equilibrium of the proposed algorithm are analyzed. The numerical results show that the performance improvement in terms of average total delay is significant (about 10 times) in comparison with “the best interface routing” especially for high throughput demands. Moreover, 20% improvement in terms of the total delay is achieved over the uniform load distribution method. This can be translated to 40% improvement in network capacity (number of CNs on the network) for the same throughput. This paper can be seen as the first stage in the dynamic flow allocation in a network of multiple cognitive users and multiple cloud servers. In our future work, we are aimed to improve the network performance while other network layers are involved in the optimization process.

References

[1] P. Loureiro, M. Liebsch, and S. Schmid, “Policy routing architecture for IP flow mobility in 3GPP’s Evolved Packet Core,” GLOBECOM Workshops, 2010 IEEE, Issue Date: 6-10 Dec. 2010, PP, 2000 - 2005.
[2] S. Aust, P. Davis, A. Yamaguchi, and S. Obana, “Interface status monitoring for wireless link aggregation in cognitive networks global,” Telecommunications Conference, GLOBECOM ’07. IEEE, Issue Date: 26-30 Nov. 2007, PP: 4873 - 4877.
[3] http://www.3gpp.org/ftp/Specs/html-info/23261.htm.
[4] Y. A. Korilis, A. A. Lazar, and A. Orda, “Achieving network optima using Stackelberg routing strategies,” IEEE/ACM Transactions on Networking, vol. 5, no. 1, pp. 161-173, February 1997.
[5] A. Orda, R. Rom, and N. Shimkin, “Competitive routing in multi-user environments,” IEEE/ACM Transactions on Networking, vol. 1, pp. 510-521, October 1993.
[6] Richard T.B. Ma and Vishal Misra,“Congestion equilibrium for differentiated service classes,” Allerton Conference on Communication, Control and Computing, September, 2011.
[7] Bertsekas, D. P. and Gallager, R. G., Data Networks, Prentice-Hall, Edition 2, Dec. 1991.
[8] D. Fundergerg and J. Tirole, Game Theory. MIT Press, 1991.
[9] D. ONeill, A.J. Goldsmith, and S.P. Boyd, “wireless Network Utility Maximization,” Military Communications Conference (MILCOM), 2008, San Diego, CA.
[10] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge University Press, 2004.
[11] J.B. Rosen, “Existence and uniqueness of equilibrium points for concave n-person games,” Econometrica, vol. 33, pp. 520-534,1965.
[12] G. Scutari, D.P. Palomar, and S. Barbarossa, “Optimal linear precoding strategies for wideband non-cooperative systems based on game theory - Part2 :Algorithms,” IEEE Trans. On Signal Processing, vol 56, no. 3, pp 1230-1249, March 2008.
[13] W. Yu, “Multi-user water-filling in the presence of crosstalk,” Information Theory and Application Workshop, pp 414-420, Feb. 2007.
[14] M.Allman, C.Hayes, H.Kruse, S.Ostermann, “TCP Performance Over Satellite Links,” Proceedings of the Fifth International Conference on Telecommunications Systems, Nashville, TN, March, 1997.