An Energy-Aware Distributed Approach for Content and Network Management

Luca Chiaraviglio
Dipartimento di Elettronica
Politecnico di Torino - Italy
Email: chiaraviglio@tlc.polito.it

Ibrahim Matta
Computer Science Department
Boston University - USA
Email: matta@cs.bu.edu

***Abstract*—We propose a distributed approach in which an Internet Service Provider (ISP) and a Content Provider (CP) *cooperate* to minimize total power consumption. Our solution is distributed between the ISP and the CP to limit shared information, such as network topology and servers' load. In particular, we adopt a dual decomposition technique. We investigate the performance of the proposed solution on realistic case-studies. We compare our algorithms with a centralized model, whose aim is to minimize total power consumption. We consider different power models for devices. Results show that the distributed algorithm is close to the optimal solution, with a power efficiency loss less than 17%.**

I. Introduction

Energy-efficient communication has become a challenging problem in the last few years. Current estimates [1] show that the Information and Communication Technology sector (ICT) consumes between 2% and 10% of the worldwide energy consumption, and this trend is expected to grow even more in the future due to the proliferation of both networked and networking devices. Internet Service Providers (ISP), are becoming sensitive to reducing the power consumption of their infrastructure, due to increasing energy costs and new business opportunities that can be realized by "going green". At the same time, Content Providers (CP) are faced with a constant increase in the number of users coupled with the need to reducing the energy consumption of both server farms and cooling systems. Therefore, both ISPs and CPs could potentially realize great benefits if energy-efficient techniques would be fully developed for network devices [2], [3] and servers [4], [5].

In this work, we propose a new approach to reducing power consumption for ISPs *and* CPs. In particular, we solve a multi-objective problem in which a CP and an ISP *cooperate* to reduce overall power consumption. We assume that the ISP is the owner of a network infrastructure. Additionally, we represent the CP infrastructure as a set of servers placed in different cities. We assume that users request content from the CP. Then, we aim at controlling the whole system composed of the ISP and the CP in order to find the minimal set of network resources and servers that minimize the total power consumption while satisfying the current content requests.

Previous works in the literature have considered the minimization of power consumption either for ISPs [6] or CPs [7] alone, thus completely ignoring the impact of one provider on the other. Considering jointly ISP and CP power consumption seems to be a viable solution to globally reduce power consumption.

In [8] we have shown that a cooperative approach is crucial to minimize overall power consumption of CP and ISP, and that up to 80% of power is wasted if either a CP-only or an ISP-only power optimization is adopted. Intuitively, a CP-only approach may result in choosing a more energy-efficient server but so much more distant that more power is consumed over the network. Conversely, an ISP-only approach may yield a server that is closer to the terminal (client) but whose power cost is high. Thus, a joint power optimization provides the right balance and yields higher power savings.

While the results in [8] show that large energy-savings are possible, it also true that nowadays CPs and ISPs are not willing to cooperate, resulting in high inefficiencies. Therefore, incentives should be proposed by third-party authorities, such as a national government, to induce cooperation between CPs and ISPs. We recognize however that avoiding sharing information is crucial for both ISPs and CPs. Therefore in this paper, differently from [8], we propose and solve a distributed approach to minimize power consumption while limiting the amount of shared information, such as the network topology and the servers' load. Moreover, we consider explicitly the algorithms from a more practical point of view, investigating the performance tradeoffs. Finally, we consider different functions to model power consumption of devices, showing that near-optimal solutions are achievable in all cases.

The paper is organized as follows. Sec. II introduces the problem and the notations. Sec. III details the proposed distributed algorithm. Simulation results are presented in Sec. IV. Finally, conclusions are drawn in Sec. V.

II. Problem Definition

The problem: The main goal of our approach is to minimize power consumption jointly between the CP and the ISP. In particular, we assume that the ISP manages a physical topology, i.e. a set of nodes and links. The CP is composed of a number of servers connected to the edge nodes of the ISP. When a user asks for a CP's resource, we assume that the resource is replicated over the CP infrastructure, so that the user can be potentially served by *any* of the servers of the CP. In a real Content Delivery Network, content is replicated according to its popularity: less popular content is normally replicated only on a subset of servers, while very popular content may be replicated *everywhere*. In this work we assume that users request only very popular content so that the effect of less popular content is negligible. Henceforth, we use the terms “node” and “router” interchangeably. Similarily, we interchangeably use the terms “terminal”, “user” and “client” to identify edge nodes of the ISP.

Basic Notations: More formally, we represent the ISP topology as a di-graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. Vertices represent network nodes, while edges represent network links. Let C_l be the capacity of link l, and let $U_l^{MAX} \in [0, 1]$ be the maximum link utilization that can be tolerated.[1] S is the set of servers of the content provider. Denote by W_s the maximum load allowed on server $s \in S$. Let R_t be the traffic demand between terminal $t \in T$ and the content provider S. Moreover, let x^{st} be real-valued variables representing the amount of traffic between a source node s and a terminal t. We divide x^{st} into x_m^{st} and x_b^{st} to denote the amount of traffic originating from the content provider under consideration and from other content providers, respectively. Actually, x_b^{st} are constants, i.e. the considered CP can not modify these traffic demands. On the other hand, we assume that x_m^{st} are real-valued variables so that a traffic demand R_t from terminal t can be served by any of the $s \in S$ CP servers, while satisfying load and delay constraints. Finally, D_{MAX} represents the maximum admissible delay.[2]

We now introduce the network-related variables. Let δ_{lp}^{st} be constants which take the values of 1 if link l belongs to path p carrying demand from s to t, 0 otherwise. Let z_p^{st} and q_p^{st} be real-valued variables representing the amount of traffic from s to t on path p for the considered CP and for other CPs, respectively. Let $\mathcal{P}(s, t)$ be the set of pre-computed paths from s to t. Additionally, let f_l be the total amount of flow on link l. Let d_l be the delay on link l, which can be approximated as a piecewise linear function of f_l, as done in [8].

Finally, we assume that the power consumption of each device (either a network node, a link or a server) depends on its actual load.

More formally, we define the monotonically increasing convex functions $P_l^d(f_l)$, $P_n^d(\sum_{l\in\mathcal{L}(n)} f_l)$, $P_s^d(\sum_{t\in T} x_m^{st})$ representing the dynamic power consumption of link l, node n and server s, respectively. $\mathcal{L}(n)$ denotes the set of links incident to node n. We then define the terms $P_n^c y_n$ and $P_s^c y_s$, which represent the startup power consumed by node n and server s when powered on. y_n and y_s are binary variables which take the value of 1 if node n and server s are powered on, respectively.

Formulation: Starting from the model presented in [8], we first define an equivalent centralized model by introducing the estimated demands $\widetilde{x}_m^{st}$ and the estimated delay $\widetilde{d}_a$ as additional variables. We then define the green centralized problem (G) as follows:

$$\textbf{G} \quad \min\left(P_{TOT} = P_{CP} + P_{ISP}\right) \quad \text{s.t.:}$$

$$P_{ISP} = \sum_{l\in E} P_l^d(f_l) + \sum_{n\in V}\left[P_n^d\left(f_{l\in\mathcal{L}(n)}\right) + P_n^c y_n\right] \tag{1}$$

$$\sum_{s\in S} \widetilde{x}_m^{st} = \widetilde{R}_t \quad \forall t \in T \tag{2}$$

$$\sum_{p\in\mathcal{P}(s,t)} q_p^{st} = x_b^{st} \quad \forall s, t \tag{3}$$

$$\widetilde{x}_m^{st} = \sum_{p\in\mathcal{P}(s,t)} z_p^{st} \quad \forall s, t \tag{4}$$

$$f_l = \sum_{s,t,p\in\mathcal{P}(s,t)} \left[\delta_{lp}^{st} z_p^{st} + \delta_{lp}^{st} q_p^{st}\right] \le C_l U_l^{MAX} \ \forall l \tag{5}$$

$$d_l \ge a_i f_l + b_i \quad \forall l, i \in I \tag{6}$$

$$d_a = \frac{\sum_l d_l}{|T|} \tag{7}$$

$$\sum_{l\in\mathcal{L}(n)} f_l \le M_n y_n \quad \forall n \in V \tag{8}$$

$$P_{CP} = \sum_{s\in S}\left[P_s^d\left(x_m^{st}\right) + P_s^c y_s\right] \tag{9}$$

$$\tag{10}$$

$$\sum_{s\in S} x_m^{st} = R_t \quad \forall t \in T \tag{11}$$

$$\widetilde{d}_a \le D^{MAX} \tag{12}$$

$$\sum_{t\in T} x_m^{st} \le W_s \quad \forall s \in S \tag{13}$$

$$\sum_{t\in T} x_m^{st} \le M_s y_s \quad \forall s \in S \tag{14}$$

$$\widetilde{d}_a = d_a \tag{15}$$

$$\widetilde{x}_m^{st} = x_m^{st} \quad \forall S \times T \tag{16}$$

Control variables: $z_p^{st} \ge 0$, $q_p^{st} \ge 0$, $y_n \in \{0, 1\}$, $y_s \in \{0, 1\}$.

The objective function of the G problem is the minimization of the total power consumption of the ISP and the CP, adopting as control variables the amount of traffic between every server-terminal pair, i.e. $z_p^{st} \ge 0$ and $q_p^{st} \ge 0$.

Considering the ISP, Eq.(1) computes its total power consumption. Eq.(2) imposes that estimated CP traffic is equal to estimated terminal demand. Notice that here we assume that $\widetilde{R}_t$ is the ISP estimation of total traffic R_t from client t. Routing constraints are specified by Eq.(3) and (4). The total flow on each link is computed and constrained by Eq.(5). Then, Eq.(6) computes the total delay for each link, using the additional variables $d_l \ge 0$. The delay function is approximated by I linear segments as in [9]. Finally, the average network delay is computed by Eq.(7).

Considering the CP, Eq.(9) computes its total power consumption. Eq.(11) guarantees the traffic demand constraint. Eq.(12) bounds the average delay of users. Eq.(13) limits the maximum load on each server.

Finally, Eq.(8) and Eq.(14) impose powering-on a network node and a server, respectively, if their incoming/outgoing

[1] Link utilization is normally kept below 100% to meet Quality of Service (QoS) requirements.

[2] We refer the reader to [8] for analysis on how the maximum admissible delay impacts power consumption.

flows are larger than zero, adopting a big-M method, i.e. $M_s \geq W_s$ and $M_n \geq \sum_{l \in \mathcal{L}(n)} C_l$ Moreover, Eq.(15) and Eq.(16) act as consistency constraints, guaranteeing that the estimated values are always equal to the real ones.

The equivalent model G belongs to the class of mixed-integer problems, that can be solved using standard optimization programs.

III. Distributed Algorithm

Two considerations hold for the G model: (i) the problem can be completely split between the ISP and the CP using a decomposition technique, (ii) after the problem is split the ISP works on the estimation of the traffic demands, while the CP uses an estimation for the users' delay.

We therefore apply the dual decomposition to derive a distributed algorithm, following a well-known procedure in the literature [10]. Here we report only the main steps; the complete description is detailed in [11]. In brief, we first introduce the Lagrange multipliers λ^{st} and μ_a associated with the consistency constraints of Eq.(15) and Eq.(16). The Lagrange multipliers are shared between the ISP and the CP. We then define the ISP problem as follows:

$$\textbf{D-GreenISP}: \quad \min\left(P_{ISP} - \lambda^{st}\widetilde{x}^{st}_m + \mu_a d_a\right) \quad \text{s.t.: (1)-(8)}$$

Control variables: $z^{st}_p \geq 0$, $q^{st}_p \geq 0$, $y_n \in \{0,1\}$.

The CP instead defines the following problem:

$$\textbf{D-GreenCP}: \quad \min\left(P_{CP} + \lambda^{st}x^{st}_m - \mu_a\widetilde{d}_a\right) \quad \text{s.t.: (9)-(14)}$$

with control variables: $x^{st}_m \geq 0$, $\widetilde{d}_a \in \mathcal{R}^+$, $y_s \in \{0,1\}$.

In order to get an optimal solution, the **D-GreenISP** and the **D-GreenCP** are solved using an iterative method that involves the Lagrange multipliers. In particular, at each iteration k the Lagrange multipliers are updated using a subgradient method:

$$\lambda^{st}(k+1) = \lambda^{st}(k) - \alpha_k\left[\widetilde{x}^{st}_m(k) - x^{st}_m(k)\right] \;\forall s,t \quad (17)$$

$$\mu_a(k+1) = \mu_a(k) - \alpha_k\left[\widetilde{d}_a(k) - d_a(k)\right] \quad (18)$$

with α_k small or diminishing step size. The intuition is that the Lagrange multipliers act as penalty/reward for the objective functions. For example, when $\widetilde{x}^{st}_m(k) - x^{st}_m(k) > 0$ the associated multiplier $\lambda^{st}(k+1)$ is decreased. When $\lambda^{st}(k+1)$ is positive, it acts as a reward for the ISP and a penalty for the CP. In our example, at iteration $k+1$ the ISP will decrease $\widetilde{x}^{st}_m(k+1)$ since the associated reward $\lambda^{st}(k+1)$ is decreased, and the CP will increase $x^{st}_m(k+1)$ since the associated penalty $\lambda^{st}(k+1)$ is decreased. Note that at equilibrium, i.e. when Eq.(15) and (16) hold, the solution of the distributed algorithm is optimal.

Since the Lagrange multipliers update needs the demands and the delays from both the ISP and the CP, we propose the adoption of a trusted third-party server (TS) to delegate the manipulation of the Lagrange multipliers. The TS can be controlled by a trusted authority that ensures that both ISPs and CPs are actively cooperating in reducing power consumption. The trusted authority incentives the cooperation between ISP and CP.[3]

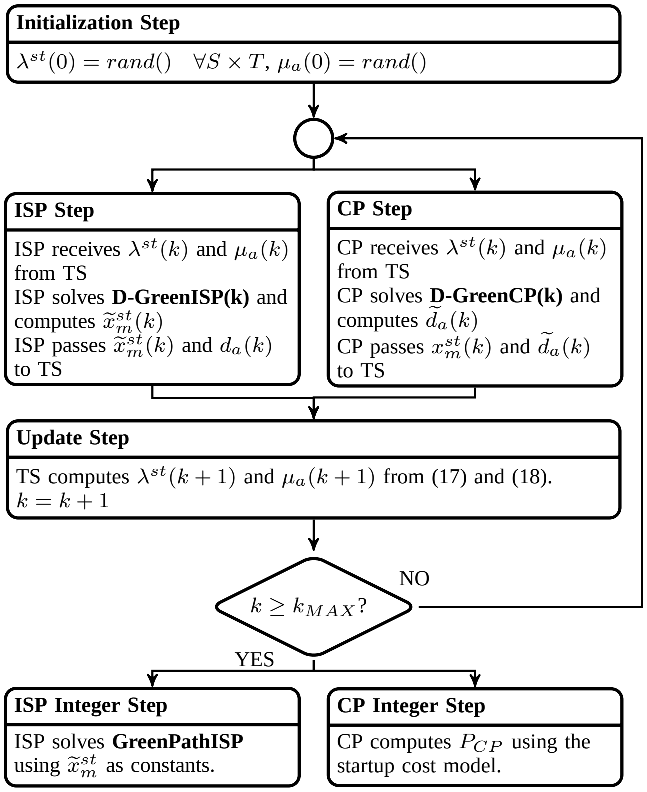
Initialization Step
$\lambda^{st}(0) = rand() \quad \forall S \times T$, $\mu_a(0) = rand()$
ISP Step
ISP receives $\lambda^{st}(k)$ and $\mu_a(k)$ from TS
ISP solves D-GreenISP(k) and computes $\widetilde{x}^{st}_m(k)$
ISP passes $\widetilde{x}^{st}_m(k)$ and $d_a(k)$ to TS
CP Step
CP receives $\lambda^{st}(k)$ and $\mu_a(k)$ from TS
CP solves D-GreenCP(k) and computes $\widetilde{d}_a(k)$
CP passes $x^{st}_m(k)$ and $\widetilde{d}_a(k)$ to TS
Update Step
TS computes $\lambda^{st}(k+1)$ and $\mu_a(k+1)$ from (17) and (18).
$k = k + 1$
$k \geq k_{MAX}$?
NO
YES
ISP Integer Step
ISP solves GreenPathISP using $\widetilde{x}^{st}_m$ as constants.
CP Integer Step
CP computes P_{CP} using the startup cost model.

Fig. 1. Dual Green algorithm (D-G).

The dual algorithm then works as follows: the Lagrange multipliers are initialized by the TS, then the **D-GreenISP** and the **D-GreenCP** are solved in parallel by the ISP and the CP, respectively, using the current Lagrange multipliers. At the end of each iteration the TS updates λ^{st} and μ_a using Eq.(17) and (18). The distributed problems are iteratively solved until a maximum number of iterations k_{MAX} is reached.[4]

Let P^G_{TOT} be the optimal total power consumption obtained from the centralized G model. Let $P^{D-G}_{TOT}(k)$ be the total power consumption at iteration k obtained from the dual algorithm. Since the subgradient method adopted in the Update Step is not a descent method, we keep track of the best distributed solution k_{Best} found so far:

$$P^{D-G}_{TOT}(k_{Best}) = \min_{i=1,..,k} P^{D-G}_{TOT}(i) \quad (19)$$

where k is the current iteration. We then define the precision error for the current solution as:

$$e_P(k) = \left|P^{D-G}_{TOT}(k) - P^G_{TOT}\right| \quad (20)$$

Similarly, we define $e_P(k_{Best})$ as the precision error considering the best distributed solution k_{Best} found so far. As

[3] Normally multiple ISPs and CPs are present. Nevertheless, our solution can be adopted also in this case. For example, a federation of trusted servers can be deployed.

[4] Another stopping criterion might be to test that $|\lambda^{st}(k+1) - \lambda^{st}(k)|$ and $|\mu_a(k+1) - \mu_a(k)|$ are very small.

reported by [12] $e_P(k_{Best})$ depends mainly on α_k. In particular, if the associated Lagrangian is a continuous function and a diminishing step size rule is adopted for α_k, then $e_P(k_{Best}) \rightarrow 0$ as $k \rightarrow \infty$, i.e. the distributed solution converges to optimum. However, in our case the Lagrangian associated with the G model is not continuous, due to the presence of the integer variables y_n and y_s. Therefore, the resulting distributed problem does not converge to an optimal solution [13]. Additionally, the distributed problem does not even converge to an equilibrium point, since the consistency constraints are not assured by the distributed approach. This impacts negatively the QoS of users, because traffic demands and delays are not properly estimated.

To overcome these problems, we propose to solve the distributed solution using only the continuous part of the power functions, then we add the integer variables locally at a second step. We name this algorithm Dual Green (D-G). In particular, the ISP and the CP solve initially the distributed step of D-G, representing the power consumption as a convex function, i.e. $y_n = 0 \; \forall n$, $y_s = 0 \; \forall s$. After few iterations, the problem converges to an optimal solution, for which both $\widetilde{x}_m^{st}$ and $\widetilde{d}_a$ are correctly estimated. Interestingly, at the end of this step both the ISP and the CP have agreed on a possible feasible solution. Then as a second step, the ISP optimizes the power consumption using the integer variables y_n and the estimated traffic demand $\widetilde{x}_m^{st}$ computed in the first step, as follows:

$$\textbf{GreenPathISP} \quad \min\left(P_{ISP}\right) \quad \text{s.t.:}$$

$$(1), (3) \tag{21}$$

$$\sum_{p \in \mathcal{P}(s,t)} z_p^{st} \;=\; \widetilde{x}_m^{st} \quad \forall s, t \tag{22}$$

$$(5) - (8) \tag{23}$$

Control variables: $z_p^{st} \geq 0$, $q_p^{st} \geq 0$, $y_n \in \{0, 1\}$.

Notice that the **GreenPathISP** problem optimizes the power consumption over the set of paths taking as inputs the traffic demands $\widetilde{x}_m^{st}$.

In parallel, the CP computes its power consumption from the demands x_m^{st} using Eq.(9), setting $y_s = 1$ for active CP servers. Fig. 1 shows a schematic description of the D-G algorithm.

IV. Performance Evaluation

We test the effectiveness of the proposed algorithms using ISP backbone topologies obtained from RocketFuel [14]. We consider the case in which the CP infrastructure is composed of 15 servers, placing the servers in the cities with the highest connection degree. We use the same set of parameters of [8]: for each (s, t) we compute up to two completely disjoint paths,[5] C_l is set to 10 Gbps for each link, $U_l^{MAX} = 0.5 \quad \forall l \in E$, to avoid congestion and to guarantee Quality of Service (QoS). The CP traffic demand R_t is modeled according to a Pareto distribution, with a variable lower bound R_t^{min} and a constant upper bound R_t^{MAX} given by the total capacity offered at that node, i.e. $R_t^{MAX} = \sum_{l \in \mathcal{L}(t)} C_l U_l^{MAX}$. Unless otherwise specified, $D^{MAX} = 300$ ms and $R_t^{min} = 10$ Mbps. We assume that nodes are connected by optical links, in which the optical carrier is regenerated by amplifiers. For each link we randomly assign a number of amplifiers A_l uniformly distributed between 1 and 5.

TABLE I
Power Consumption Model [W]

P_n^c	$P_l^d(f_l)$	$P_n^d\left(\sum_{l \in \mathcal{L}(n)} f_l\right)$	P_s^c	$P_s^d\left(\sum_{t \in T} x_{cp}^{st}\right)$
100	$20 f_l A_l$	$20 \sum_{l \in \mathcal{L}(n)} f_l$	200 ± 100	$(40 \pm 20) \sum_{t \in T} x_{cp}^{st}$

Tab.I describes the model used to evaluate the power consumption. Here we are assuming next-generation devices able to adapt their power with traffic flow [2]. Considering the ISP, the power consumption of nodes is composed of a constant term P_n^c due to the chassis static power plus an additional term P_n^d which scales linearly with traffic flow. Moreover, the power consumption of a link P_l^d depends linearly on both the load and the number of amplifiers A_l between nodes. The constant values are extracted by interpolating the power measurements of real devices under high load [8].

Focusing on CP, the server power consumption is also modeled by a static term P_s^c and a dynamic term P_s^d: in this case instead the slope is higher due to the presence of backup elements and power supplies, which actually double the server power consumption. For the sake of simplicity we do not consider any additional background traffic of other CPs, since our goal is mainly to assess the maximum power savings achievable by the whole system composed of the ISP and the considered CP. Finally, 50% of randomly chosen nodes are selected as terminals t.

A. Algorithm Comparison

We start by running the D-G algorithm over the SprintLink topology, since it is one of the largest topologies of RocketFuel in terms of nodes and links. Unless otherwise specified, we assume that the ISP knows exactly the total traffic of each client, i.e. $\widetilde{R}_t = R_t$.[6] Moreover, we use the power model presented in Tab.I. We name this power model as "100-200". Finally, we set a step size rule $\alpha_k = 1000/k$ for updating the Lagrange multipliers.

Fig. 2-(a) reports the power consumption variation of the D-G algorithm versus the number of iterations, considering the 100-200 model. Notice that here, differently from the original scheme of Fig. 1, we perform the ISP and CP Integer Step at each iteration to better assess the dynamic behavior of the algorithm. We report also a lower bound, i.e. the power consumption of the G algorithm. The figure reports also an upper bound, namely the classic (C) centralized solution presented in [8], whose objective is to minimize the users' delay. Finally, the figure reports the power consumption of the D-G algorithm at the end of the distributed step, before **GreenPathISP** is solved. Several considerations hold in this

[5] As in [8] the topologies are first pre-processed using a simple shortest path algorithm to obtain the set of paths. The set of paths is given as input to our optimization problems. This reflects normal behavior of ISP that selects alternate paths for failure protection.

[6] $\widetilde{R}_t$ is measured or computed from previous estimations of the traffic demands.

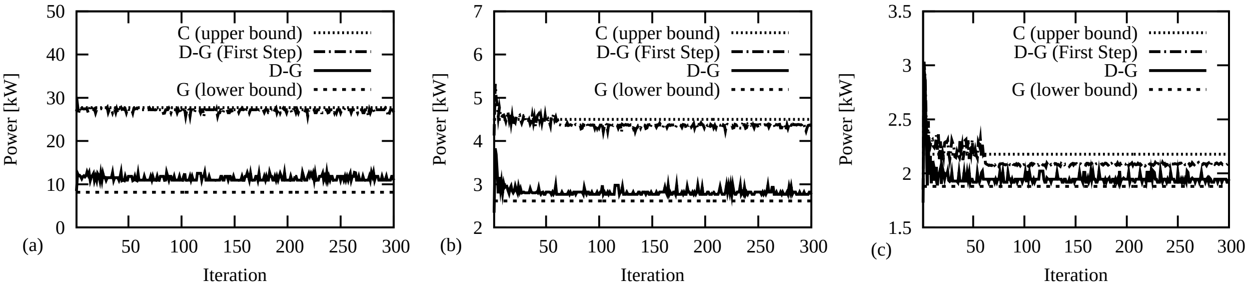

Fig. 2. Power consumption of the C, G and D-G algorithms: **(a)** 100-200 power model, **(b)** 10-20 power model, **(c)** 1-2 power model.

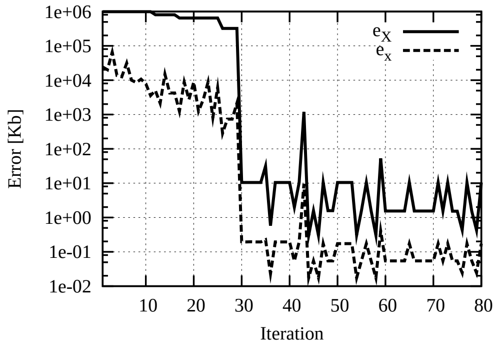

Fig. 3. e_x and e_X for the SprintLink topology.

case: (i) the solution of D-G is always close to the lower bound even after few iterations, so that the maximum power loss is less than 17%,[7] (ii) the power consumption of the first step of D-G is instead close to the upper bound, (iii) the optimization performed by **GreenPathISP** is essential to obtain large savings, since P_{TOT} drops from more than 26 kW in the first step to less than 13 kW at the end of the algorithm.

We then investigate how the startup cost impacts the total power consumption. Fig. 2-(b) shows the results for the 10-20 model, i.e. $P_n^c = 10$ W and $P_s^c = 20 \pm 10$ W. As expected, the power consumption of the G algorithm is lower in this case, and the bounds are closer too. Interestingly, the D-G is even closer to the lower bound, since the linear part of the power function that is optimized in the Distributed Step becomes predominant. These phenomena are even more evident with the 1-2 model (Fig. 2-(c)): in this case the upper and lower bound are even closer, suggesting that with small power step sizes the solution of the D-G algorithm approaches that of the G one.

We define the mean error of the traffic demands at each iteration:

$$e_x(k) = \frac{\sum_{s,t} |\widetilde{x}_m^{st}(k) - x_m^{st}(k)|}{|S \times T|} \tag{24}$$

In a similar way we define the maximum error at each iteration:

$$e_X(k) = \max_{s,t} \left|\widetilde{x}_m^{st}(k) - x_m^{st}(k)\right| \tag{25}$$

Notice that when the algorithm converges $e_X \approx 0$, i.e. $\widetilde{x}_m^{st} \approx x_m^{st} \quad \forall s,t$.

Fig. 3 reports both e_x and e_X at each iteration. Interestingly, e_x falls below 100 Kb after 30 iterations, while e_X is bounded below 100 Kb after 45 iterations. This means that only few iterations are sufficient to guarantee QoS for users, since the estimated demands $\widetilde{x}_m^{st}$ are close to the real ones x_m^{st}.

B. Parameter Impact

We then consider the impact of parameters on the performance of the algorithm. In particular, we start considering the case in which the power consumption is strictly proportional with the current load. This case can be representative of future energy-aware devices, able to completely adapt the power consumption to the current load [2]. Fig. 4 (left) reports the precision error $e_P(k_{Best})$ for $k \in [1, 300]$, considering different diminishing step size rules for α_k. We set $k_{max} = 300$ to limit the convergence time. Small step sizes lead to very slow convergence, since the Lagrange multipliers change very slowly. For example, with $\alpha_k = 10/k$ the error is always higher than 9%, meaning that the distributed solution is quite far from the centralized one. However, also large values tend to be inaccurate since large oscillations are induced. By choosing instead the intermediate value of $1000/k$, the D-G algorithm converges to the optimal solution in less than 50 iterations with a precision of less than 0.0001%.

Fig. 4 (center and right) show the power consumption of the ISP and CP, respectively. Interestingly, all the step sizes are able to reach at least a near-optimal power consumption for the ISP, being $1000/k$ and $1000/k^{1/2}$ the noisiest ones due to the large steps used. If we consider instead the CP power consumption, then only when α is greater than $100/k^{1/2}$, the CP converges to the optimal power allocation, while all the other values are quite far from the optimal solution.

We then extend our analysis to other ISP topologies [14]. Due to lack of space we refer the reader to [11] for these results. In brief, the $1000/k$ rule is able to achieve a minimum precision of 0.001% for all the topologies considered in less than 50 iterations.

To better assess the computational time of D-G, we compute the CPU time required to solve the problem at each iteration. In particular, since the ISP Step and the CP Step can be processed in parallel, we take the maximum of the CPU times: $ctime(k) = \max(ctime_{ISP}(k), ctime_{CP}(k))$.

[7] The power loss is computed as the difference between D-G and GS in terms of power savings, i.e. $P_{loss} = \frac{P_{TOT}^{D-G} - P_{TOT}^{G}}{P_{TOT}^{C}}$.

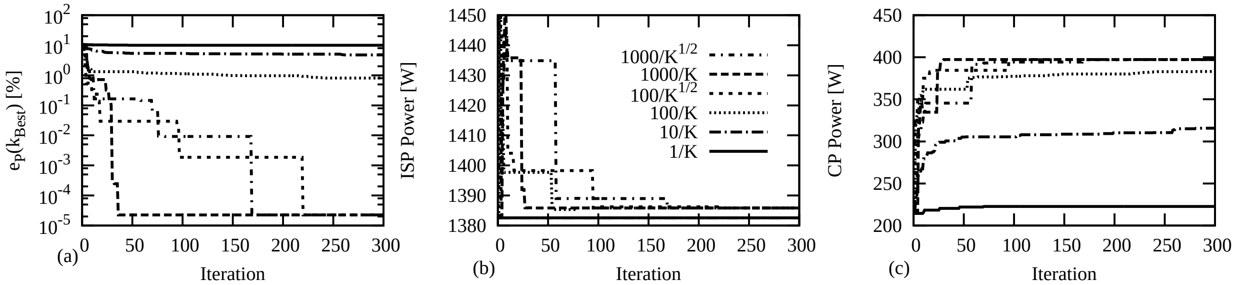

Fig. 4. Impact of different decreasing α_k considering the SprintLink topology: **(a)** $e_P(k_{Best})$, **(b)** ISP power consumption, **(c)** CP power consumption.

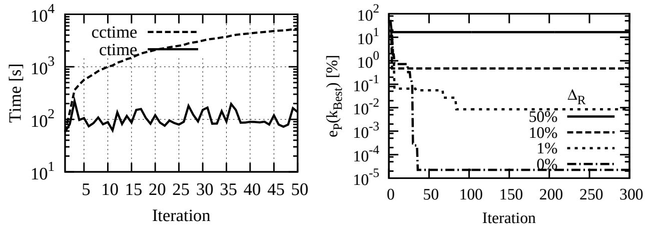

Fig. 5. (left) $ctime(k)$ and $cctime(k)$ of D-G considering the SprintLink topology and $1000/k$ step size rule. (right) $e_P(k_{Best})$ for different precision values Δ_R.

We then compute the total cost of running the algorithm at iteration k as $cctime(k) = cctime(k-1) + ctime(k)$, where $cctime(1) = ctime(1)$. We assume that the CPU times required to perform the Initialization and the Update steps of D-G are negligible.

Fig. 5(left) reports $ctime(k)$ and $cctime(k)$ for the SprintLink topology and $1000/k$ step size rule. All the times have been measured by running D-G on the NEOS server [15] to obtain a reliable measurement on a widely known system. Interestingly, $ctime(k)$ is nearly constant, so that 30 iterations require 52 minutes to be completed. Clearly, a tradeoff emerges between solution precision and admissible computational time.

Finally, we introduce a precision error for $\widetilde{R}_t$, so that $\widetilde{R}_t = R_t(1 + \Delta_R)$. This reflects the case in which R_t is over-estimated by the ISP, for example by measurements.[8] Fig. 5(right) reports $e_P(k_{Best})$ for different Δ_R. For $\Delta_R = 0\%$ and $\Delta_R = 1\%$ the algorithm converges to the optimal power consumption and $e_P(k_{Best})$ falls below 0.01% in less than 90 iterations in both cases. For $\Delta_R = 10\%$ instead the D-G algorithm requires 0.46% of additional power than the G algorithm, which rises to more than 16% with $\Delta_R = 50\%$ even after 300 iterations. Therefore, the solution produced by the D-G algorithm involves a small amount of additional power only when the precision error in the estimated demand is reasonably small.

V. Conclusions and Future Work

In this work, we have proposed a distributed approach to minimize the total power consumption of an ISP and a CP. After showing that the problem can be formalized with an optimization model, we have developed a distributed algorithm based on a dual decomposition approach. Results show that the distributed solution is near-optimal for all considered scenarios, with a maximum power efficiency loss of 17%.

As future work, we intend to study further the implications on new CP-ISP architectures. We aim to consider the interaction of multiple CPs over multiple ISPs to minimize power consumption, considering also the effects of server virtualization on potential power savings possible with colocating CPs. Then, we want to assess through simulation how the size of the CP and its location in the ISP network impact the performance of the distributed algorithm. Finally, we will evaluate our solutions in the face of temporal variations in traffic, considering the tradeoffs between the precision of the solution and the maximum computational time.

References

[1] SMART 2020, *Enabling the Low Carbon Economy in the Information Age*, http://www.theclimategroup.org, 2009.

[2] R. Bolla, R. Bruschi, K. Christensen, F. Cucchietti, F. Davoli; S. Singh, "The Potential Impact of Green Technologies in Next-Generation Wireline Networks – Is There Room for Energy Saving Optimization?," *IEEE Communication Magazine (COMMAG)*, 2011.

[3] F. Idzikowski, S. Orlowski, Ch. Raack, H. Woesner, and A. Wolisz, "Saving energy in IP-over-WDM networks by switching off line cards in low-demand scenarios", *ONDM '10*, Kyoto, Japan, February 2010.

[4] D. Meisner, B. Gold, and T. Wenisch, "PowerNap: Eliminating Server Idle Power," *ASPLOS '09*, Washington DC, USA, March 2009.

[5] L.A. Barroso, and U. Holzle, "The Case for Energy-Proportional Computing," *IEEE Computer*, Jan 2007.

[6] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power Awareness in Network Design and Routing," *IEEE INFOCOM*, Phoenix, AZ, USA, April 2008.

[7] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs., "Cutting the Electric Bill for Internet-Scale Systems," *ACM SIGCOMM*, Barcelona, Spain, August 2009.

[8] L. Chiaraviglio and I. Matta, "GreenCoop: Cooperative Green Routing with Energy-efficient Servers," *ACM e-Energy 2010*, Passau, Germany, April 2010.

[9] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang, "Cooperative Content Distribution and Traffic Engineering in an ISP Network," *SIGMETRICS/Performance*, Seattle, WA, USA, June 2009.

[10] S.P. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, UK, 2004.

[11] L. Chiaraviglio and I. Matta, "A Green Distributed Cooperation for Network and Content Management," *Technical Report*, CS Department, Boston University, MA, USA, March 2010.

[12] L. Xiao, M. Johansson, and S.P. Boyd, "Simultaneous Routing and Resource Allocation via Dual Decomposition," *IEEE Transactions on Communications*, Volume 52, Number 7, 1136-1144, 2004.

[13] A.J. Conejo, E. Castillo, R. Minguez, and R. Garcia-Bertrand, *Decomposition techniques in mathematical programming: engineering and science applications*, Springer Verlag, New York, NY, USA, 2006.

[14] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP Topologies with Rocketfuel," *ACM SIGCOMM*, Pittsburgh, PA, USA, August 2002.

[15] *NEOS Server for Optimization*, http://neos.mcs.anl.gov/neos/.

[8] We do not consider the under-estimated case since it introduces packet-loss and consequently QoS violation for users.