Greening the Internet: Energy-Optimal File Distribution

Kshitiz Verma*[†], Gianluca Rizzo*, Antonio Fernández Anta*, Rubén Cuevas Rumín[†], Arturo Azcorra[†]
{kshitiz.verma, gianluca.rizzo, antonio.fernandez}@imdea.org, {rcuevas, aa}@it.uc3m.es
[†]*Universidad Carlos III de Madrid, Spain*
Institute IMDEA Networks, Madrid, Spain

Abstract—**Despite file distribution applications are responsible for a major portion of the current Internet traffic, so far little effort has been dedicated to study file distribution from the point of view of energy efficiency. In this paper, we present the first extensive and detailed theoretical study for the problem of energy efficiency in file distribution. Specifically, we first demonstrate that the general problem of minimizing energy consumption in file distribution is NP-hard. For restricted versions of the problem, we derive tight lower bounds on energy consumption, and we design a family of algorithms that achieve these bounds. Our results prove that through collaborative p2p schemes up to 50% energy savings are achievable with respect to the best available centralized file distribution scheme. Through simulation, we show that even in heterogeneous settings (e.g., considering network congestion, and link variability across hosts) our collaborative algorithms always achieve significant energy savings with respect to the power consumption of centralized file distribution systems.**

I. INTRODUCTION

The need for a reduction in the carbon footprint of all human activities while satisfying an ever growing energy demand has triggered the interest on the design of novel energy-efficient solutions in multiple domains [1]. In the Information and Communication Technologies (ICT) domain, this has produced research in novel mechanisms and solutions for saving energy, to be deployed in computation and communication systems [2]–[5]. In communication systems, the proposed approaches for energy efficiency aim at achieving "energy proportional" networks, i.e., networks that consume an amount of power proportional to their traffic load. This is done through both the design of new energy-efficient hardware [6] and by acting at the system level (e.g., via energy-efficient routing [7], [8] or "sleep modes" [9], [10]). In addition, an important challenge is how to save energy in end systems (servers and user terminals) at the edge of the network, which are responsible for the major portion of the whole Internet power consumption [11], [12]. Currently, power saving technologies for end systems are available, like switching off or to low-power modes the devices when possible [12]. These techniques can be used to approach energy proportionality in these systems, i.e., making the power consumed proportional to the level of CPU or network activity.

However, energy proportionality of the different elements alone is not enough to reduce energy wastage in most distributed systems. It needs to be complemented by a redesign of the services (e.g., file sharing, web browsing, etc.) in a way that optimizes the utilization of hosts and network resources. The reason is that most common services have been designed without taking energy into account. In fact, typically the waste of network resources has been seen as an issue only when it implies significant higher cost or poor performance. Raising energy costs and increasing environmental awareness are pushing to reconsider also this aspect.

In this paper we focus on the file distribution service, which is one of the most widespread services on the Internet. Services such as peer-to-peer (p2p) file sharing, one-click-hosting (OCH), software release, etc., represent indeed a major fraction of current Internet traffic [13]–[15]. In addition, within the context of corporate/LAN networks, operations such as software updates are also file distribution processes. Despite of the importance of these services, to the best of the authors' knowledge little effort has been dedicated to understanding and achieving energy-efficiency in the context of file distribution applications. All this makes essential to deeply investigate energy-efficiency in file distribution, in order achieve a truly *Green Internet*.

Related work An important amount of effort has been dedicated to study the completion time in a file distribution process [16]–[19]. The minimization of the average finish time in P2P networks is considered in [20]–[22]. Mundinger et al. [23] present a theoretical study to derive the minimum time associated to a P2P file distribution process. However, a scheme guaranteeing a file distribution with minimum completion time does not generally lead to minimize the energy consumption. Indeed, schemes with a same distribution time may have different energy costs.

To the best of the authors' knowledge, energy consumption in file distribution processes has received little attention so far. On one hand, practical studies [5], [24]–[27] have discussed and compared the energy consumed by different content distribution architectures or protocols. However none of them relies on an analytical basis nor proposes energy-optimal algorithms, as is the case of our paper. Lachlan et al. [28] consider an instance of the problem similar to ours, but under a fluid limit model, in which the file is split into infinitesimally small blocks. For this reason, the results of the paper cannot be extended to practical settings, where block sizes must be lower bounded to keep bounded the amount of extra transmissions (and extra energy spent) due to control data (protocol overheads, etc). As we show in our paper (see Section III), the dependence on the blocks size and number of the energy consumption of a distribution scheme is non negligible in any practical scenario. The authors propose a set of external behavior specifications for a family of algorithms, providing an upper bound for the total energy they consume in the distribution process. As in their algorithms a subset of hosts (which always contains at least the server) stays on for the whole duration of the scheme, the total energy consumption of the proposed algorithms is higher with respect to the optimal values (that we define here) by at least a factor directly proportional to the power consumed by the server and to the makespan of the distribution scheme. As we show later in the paper, for such schemes the total energy consumption is up to 50% with respect to the optimal schemes we propose, depending on the specific settings.

Our contributions This paper aims to define the analytical and algorithmic basis for the design of energy-efficient file distribution protocols. For this purpose, we first model the problem of minimizing energy consumption in a file distribution process. Then, we prove that the general version of the problem is NP-hard. This leads us to define restricted versions of the problem, yet maintaining a balance between simplicity and applicability in real scenarios. For these versions, we are able to derive lower bounds on the energy required to complete the file distribution process. We also propose collaborative p2p algorithms for reducing energy consumption in the studied file distribution scenarios. These algorithms are shown to be energy-optimal in a large collection of cases. Finally, we present an empirical evaluation through simulation, that allows us to validate our analytical results and study the impact of relaxing some of the assumptions imposed in the analysis on the performance of the proposed algorithms.

Our analyses and simulations demonstrate that collaborative p2p schemes are an appropriate approach to reduce the energy consumption in a file distribution process, with respect to any centralized file distribution schemes. The simulations show that, even in scenarios for which they were not designed (e.g., considering heterogeneous energy consumption or network congestion), our collaborative p2p schemes achieve significant energy savings with respect to popular centralized file distribution systems. These savings range between 50% and two order of magnitude depending on the centralized scheme under consideration.

It should be noted that the algorithms proposed only depend on the number of hosts involved and the number of blocks of the file. Hence, given these two parameters, each host can implement the scheme in a fully distributed manner.

The rest of the paper is structured as follows. Section II provides the network and energy model along with definitions and terminology used throughout the paper. Sections III and IV present theoretical results obtained, in the form of bounds and file distributions schemes, respectively for the case in which download capacity is equal to the upload capacity, and for the case in which the download capacity is larger than the upload capacity. In Section V, we present our simulation study. Section VI concludes the paper.

II. SYSTEM MODEL, PROBLEM DEFINITION AND ASSUMPTIONS

System Model and Assumptions We consider a system of $n+1$ hosts ($n \geq 1$) that are fully connected via a wired network. One of these hosts, called the *server* and denoted by S, has initially a file of size B that it has to distribute to all the other hosts, which we call the *clients*. We assume that the file is divided into $\beta \geq 1$ blocks of equal size $s = B/\beta$. The set of hosts is denoted as $\mathcal{H} = \{S, H_0, H_1, ..., H_{n-1}\}$, and the set of blocks as $\mathcal{B} = \{b_0, b_1, ..., b_{\beta-1}\}$. We will also use in this paper a set of indexes, defined as $\mathcal{I} = \{S, 0, \ldots, n-1\}$. For simplicity of notation and presentation, we will often use an index $i \in \mathcal{I}$ to denote a host, and even talk about host i instead of host H_i (or S when $i = S$).

All the hosts in $\mathcal{H}$ can upload blocks of the file to other hosts (initially only S can do so). A client can start uploading block b_i only if it has received b_i completely. Host H_i has upload capacity u_i and download capacity d_i, for $i \in \mathcal{I}$. (Observe that the server has upload capacity u_S.) We assume that all capacities are integral. All the hosts are assumed to be identical with respect to processing speed, and to have enough memory to sustain the distribution process. No host can upload more than a block at any given time instant, but can simultaneously upload and download from other hosts. Moreover, it can simultaneously download from multiple hosts as long as the download capacity allows it. We also assume that hosts always upload at their full capacity.

We assume that time in the file distribution process is slotted. Each block transmission between hosts starts and finishes within the same slot. We assume that no host uploads to more than one host in one slot. In general, the slot duration may vary from one slot to the next. However, unless otherwise stated, we will assume during the rest of the paper that all slots have the same duration γ. Then, if the process of file distribution starts at time $t = 0$, the time interval $[0, \gamma]$ corresponds to slot $\tau = 1$ and, in general, slot τ spans the time interval $[(\tau - 1)\gamma, \tau\gamma]$. In each slot of a scheme, a host is assigned another host to serve (if any), and the set of blocks it will serve during that slot.

In this work we consider only the energy consumed by hosts during the file distribution process. We do not consider the energy consumed by other network devices. In our model, the energy consumption has the following three components:

(1) Each host $i \in \mathcal{I}$, just for being on, consumes power P_i (when a host is off, we assume that it consumes no power).
(2) Each host consumes a fixed amount of energy $\delta_i \geq 0$, $i \in \mathcal{I}$ for each block served and/or received. This component δ_i captures the additional energy consumed by serving and receiving in the form of CPU activity [29], cooling, caching and hard disk activity, network card activity, etc.
(3) A host consumes energy while being switched on or off. If host $i \in \mathcal{I}$ takes time α_i to switch on or off, the energy consumed by switching is $P_i \cdot \alpha_i$. Usually, this on/off time α_i is in the order of a few seconds [30].

The Problem and its Complexity We define a *file distribution scheme*, or *scheme* for short, as a schedule of block transfers between hosts such that, after all the transfers, all the hosts have the whole file. Observe that a scheme must respect the model previously defined. Then, the problem we study in this paper is defined as follows.

Definition 1: The *file distribution energy minimization problem* is the problem of finding or designing a file distribution scheme that minimizes the total energy consumed. The bad news is that this problem is NP-hard even if switching on and off is free and there is no additional energy consumption per block. The following theorem summarizes the result. (Please refer to [31] for the proof of the theorem.)

Theorem 1: Assume that time is slotted, that hosts must upload at their full capacity, and that no host can upload to more than one host in the same slot. The problem of minimizing the energy of file distribution is NP-hard if hosts can have different upload capacities and power consumptions, even if $\alpha_i = \delta_i = 0, \forall i \in \mathcal{I}$.

The good news is that, by making a few simplifying yet realistic assumptions, we can solve the file distribution energy minimization problem optimally (Theorem 4, Section III) or near optimally (Theorem 7, Section IV).

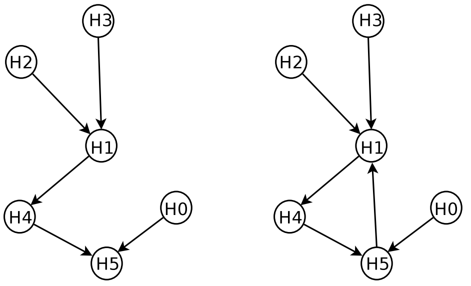
Simple Model Henceforth, we assume that all the hosts have the same upload capacity u, and the same download capacity d. We also assume that $\frac{d}{u} = k$ for some positive integer k. Unless otherwise stated, we assume that hosts are switched on and off instantaneously, i.e., $\alpha_i = 0, \forall i$, and hence switching consumes no energy.

The uniformity of capacities results in a uniform slot duration, equal to $\gamma = \frac{s}{u}$, for all the block transfers. A host is said to be *active* in a time slot if it is receiving or serving blocks in the slot. Otherwise, it is said to be *idle*. The energy Δ_i consumed by an active host $i \in \mathcal{I}$ in one slot can be computed as follows.

$$\Delta_i = P_i \gamma + \delta_i = \frac{P_i s}{u} + \delta_i = \frac{P_i B}{u\beta} + \delta_i. \tag{1}$$

Without loss of generality, we assume that $\Delta_0 \leq \cdots \leq \Delta_{n-1}$.

In some cases below we will assume that the system is *energy-homogeneous*. This means that all hosts have the same energy consumption parameters, i.e., $P_i = P$ and $\delta_i = \delta$, for all $i \in \mathcal{I}$. In such a homogeneous system, also all hosts have the same value of $\Delta_i = \Delta$. Note that, unless otherwise stated, we assume a heterogeneous system.



(a) Tree slot (b) Slot with cycle

Figure 1. A slot as a directed transfer graph. The number of blocks served in 1(b) is one more than the number of blocks served in 1(a), with the same energy consumption.

Let us consider parameters n, k, and β of the file distribution energy minimization problem. Let us define the set of all possible schemes with these parameters by $\mathcal{Z}_k^{n,\beta}$. Let $E(z)$ be the energy consumed by scheme $z \in \mathcal{Z}_k^{n,\beta}$. A scheme $z_0 \in \mathcal{Z}_k^{n,\beta}$ is *energy optimal* (or optimal for short) if $E(z_0) \leq E(z), \forall z \in \mathcal{Z}_k^{n,\beta}$. Hence, our objective in the rest of the paper is to find optimal (or quasi-optimal) schemes.

Normal Schemes To rule out redundant and uninteresting schemes, we will consider only what we call normal schemes. Observe that the block transfers of a scheme z in a slot τ can be modeled as a directed *transfer graph* with the hosts as vertices and block transfers as edges (see Fig. 1). Then, a *normal scheme* is a distribution scheme in which there are no idle hosts, there are no slots without active hosts, and each slot has exactly one connected transfer graph. We denote the set of normal schemes with parameters n, β, and k by $\hat{\mathcal{Z}}_k^{n,\beta}$. From now onwards, we will consider only normal schemes. It is easy to observe that any optimal scheme can be transformed into a normal scheme that is also optimal. Hence, we are not losing anything by concentrating only on normal ones.

Observe that in a transfer graph the out-degree of each vertex is at most 1 (by the upload constraint). Thus, the transfer graph of a slot in a normal scheme can either be a tree (Fig. 1(a)) or a graph with exactly one cycle (Fig. 1(b)). Note also that in a slot with one cycle all hosts upload blocks, while in a tree slot there is exactly one host that does not upload.

Costs Let us consider scheme $z \in \hat{\mathcal{Z}}_k^{n,\beta}$. Denote with $\mathcal{I}_\tau^z \subseteq \mathcal{I}$ the indexes of the set of active hosts in time slot τ under scheme z. Let τ_f^z be the makespan of scheme z, i.e., the time slot of z in which the distribution of the file is completed.

Definition 2: The *cost of slot* τ under scheme z, denoted c_τ^z, is the energy consumed by all active hosts $\mathcal{I}_\tau^z$ in τ, i.e, $c_\tau^z = \sum_{i \in \mathcal{I}_\tau^z} \Delta_i$, and the energy consumed by the scheme z

is $E(z) = \sum_{\tau=1}^{\tau_f^z} c_\tau^z = \sum_{\tau=1}^{\tau_f^z} \sum_{i\in\mathcal{I}_\tau^z} \Delta_i$.

The cost of a slot, as defined above, does not take into account which host is serving which block to which host. However, the total energy consumption of a scheme also depends on this. Thus, for a better insight on the schemes, we also associate a cost to a block transfer. The cost of block transfers will be used in the proofs of lower bounds. Let us denote the set of blocks downloaded by host $i \in \mathcal{I}$ in slot τ under scheme z by $\mathcal{S}_{i,\tau}^z$ and the index of the host serving $b_j \in \mathcal{S}_{i,\tau}^z$ as $serv(j,i)$.

Definition 3: We define the *cost* $c_{j,i}^z$ *of a block* b_j received by H_i under scheme z as, $c_{j,i}^z = \mathcal{D}_{j,i}^z \cdot \Delta_i + \mathcal{U}_{j,i}^z \cdot \Delta_{serv(j,i)}$, where, if b_j is received by H_i in slot τ,

$$\mathcal{D}_{j,i}^z = \begin{cases} 1 & \text{if } j = \min\{j' | b_{j'} \in \mathcal{S}_{i,\tau}^z\} \\ 0 & \text{Otherwise} \end{cases}$$

$$\mathcal{U}_{j,i}^z = \begin{cases} 1 & \text{if } \mathcal{S}_{serv(j,i),\tau}^z = \emptyset \\ 0 & \text{Otherwise} \end{cases}$$

$\mathcal{D}_{j,i}^z$ accounts for the energy consumption of host H_i (in units of Δ_i) that is receiving the block. A block contributes to the energy consumed by H_i if it is downloading. If a host is downloading more than one block in parallel, then we assume that only *one block* adds to the cost, as the rest of the blocks can be received without incurring any further cost. $\mathcal{U}_{j,i}^z$ accounts for the energy consumption of the host that is serving the block when $\mathcal{S}_{serv(j,i),\tau}^z = \emptyset$ (the host that is serving b_j to H_i is not downloading any block).

With the above definition, the sum of the costs of all blocks transferred in slot τ should be equal to the cost of the slot τ, c_τ^z. The next result establishes that this is indeed true for all the schemes. The proof can be found in [31].

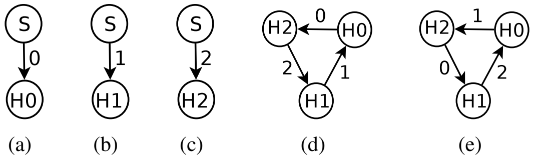
Theorem 2: The sum of the costs of all the blocks transferred during slot τ is equal to the cost of that slot, i.e., $\sum_{i\in\mathcal{I}_\tau^z} \sum_{b_j\in\mathcal{S}_{i,\tau}^z} c_{j,i}^z = c_\tau^z$. Hence, the energy consumed by the scheme is

$$E(z) = \sum_{i=0}^{n-1} \sum_{j=0}^{\beta-1} c_{j,i}^z = \sum_{i=0}^{n-1} \sum_{j=0}^{\beta-1} \left(\Delta_i \cdot \mathcal{D}_{j,i}^z + \Delta_{serv(j,i)} \cdot \mathcal{U}_{j,i}^z\right) \tag{2}$$

III. DOWNLOAD CAPACITY = UPLOAD CAPACITY

In this section and the next one we provide analytical results for the file distribution energy minimization problem, under the simple model described previously. In this section we explore the case $k = 1$. In this case, because download and upload capacities are equal, a host can download at most one block during a slot. We derive lower bounds on the energy consumed by any scheme, and design optimal schemes achieving it. We also find the optimal number of blocks to be used to minimize the energy of optimal schemes in energy-homogeneous systems.

Lower Bound The following theorem provides a lower bound on the energy consumed by any distribution scheme when $k = 1$.



(a) (b) (c) (d) (e)

Figure 2. Example of Algorithm 1, for $n = 3$ and $\beta = 3$. The label on each arrow is the index of the block being served.

Theorem 3: The energy required by any scheme z to distribute a file divided into β blocks among n clients when $k = \frac{d}{u} = 1$, satisfies $E(z) \geq \beta\left(\Delta_S + \sum_{i=0}^{n-1} \Delta_i\right) + \max\{0, n-\beta\} \min\{\Delta_S, \Delta_0\}$.

The key observation behind this result is that each host has to be active for at least β slots to receive the file, whereas the server has to be active for at least β slots to upload one copy of each block among the clients. The proof of the theorem can be found in [31].

Optimal Distribution Schemes We now present optimal schemes achieving the lower bound of Theorem 3. We distinguish among three cases, depending on the relation between n and β, and we indicate the resulting schemes as Algorithms 2, 3, and 1. Note that in the pseudocode of algorithms, the transfer of block b_j from host H to host H' is expressed as $H \xrightarrow{j} H'$. While the three algorithms could be merged into one, we have chosen to present them separately for clarity.

Theorem 4: When $d = u$, Algorithms 2, 3, and 1 describe optimal distribution schemes, with energy $E(z) = \beta\left(\Delta_S + \sum_{i=0}^{n-1} \Delta_i\right) + \max\{0, n-\beta\} \min\{\Delta_S, \Delta_0\}$.

Intuition for the optimality of the algorithms: We start with Algorithm 1 (see Fig. 2), which is the simplest of the three, since it assumes that the number of clients is equal to the number of blocks. In the first n slots of the algorithm, the server uploads a distinct block of the file to each of the n clients. Since $n = \beta$, the server can upload the whole file to the clients in n slots. Then the server goes off. At this point, each host has a different block and needs to get the remaining $n-1$ blocks. Then, in each of the remaining $n-1$ slots, each client chooses another client to serve in a way that the resulting transfer graph is a cycle of the n hosts. In particular, each host i uploads the latest block it has received to host $i-1$. This process continues for the next $n-1$ slots, until all the hosts have all the blocks.

Algorithm 1 (and Fig. 2) reflects clearly the key for the optimality of the three algorithms, which is creating cycles so that all hosts that are downloading are also uploading. Algorithm 2, which assumes $n < \beta$, is more involved, but uses similar ideas as Algorithm 1. In Algorithm 3, the number of clients is larger than the number of blocks. Thus some hosts will have to upload the same block more than once. In this algorithm, once the server has served β distinct blocks, the host with the smallest energy consumption per

Algorithm 1 Optimal scheme for $\beta = n$

```
for slot $j = 0 : n-1$
    $S \xrightarrow{j} H_j$
for slot $j = n : 2n-2$
    for $i = 0 : n-1$
        $H_i \xrightarrow{(i+j) \bmod n} H_{(i-1) \bmod n}$
```

slot uploads block b_0 to those hosts without any block.

For the complete proofs of correctness and optimality, please refer to [31]. In what follows, with $Opt(n, \beta)$ we indicate the optimal algorithm corresponding to the values of n and β.

Algorithm 2 Optimal scheme for $\beta > n$

```
for slot $j = 0 : n-1$
    $S \xrightarrow{j} H_j$
for slot $j = n : \beta-1$
    $S \xrightarrow{j} H_{n-1}$
    for $i = 1 : n-1$
        $H_i \xrightarrow{i+j-n} H_{i-1}$
for slot $j = \beta : \beta+n-2$
    for $i = 1 : n$
        $H_{i \bmod n} \xrightarrow{(i+j-n) \bmod \beta} H_{i-1}$
```

Algorithm 3 Optimal scheme for $\beta < n$.
$H_{\min}$ is the host with smallest Δ_i. ($H_{\min} \in \{S, H_0\}$.)

```
for slot $j = 0 : \beta-1$
    $S \xrightarrow{j} H_j$
for slot $j = \beta : n-1$
    $H_{\min} \xrightarrow{0} H_{j+1-\beta}$
    for $i = 1 : \beta-1$
        $H_{i+j-\beta} \xrightarrow{i} H_{i+j+1-\beta}$
for slot $j = n : n+\beta-2$
    $H_{2n-(j+1)} \xrightarrow{\beta-1} H_{n+\beta-(j+2)}$
    for $i = 0 : \beta-2$
        $H_{(n+i-j) \bmod n} \xrightarrow{i} H_{(n+i-j-1) \bmod n}$
```

Optimal Number of Blocks in Homogeneous Systems Consider now an energy-homogeneous system, in which all the hosts have the same energy consumption parameters, i.e., $P_i = P$ and $\delta_i = \delta$, for all $i \in \mathcal{I}$. Our goal is to find the optimal value of β into which the file should be divided for minimum energy consumption. The following theorem summarizes the result.

Theorem 5: In an energy-homogeneous system with $k = \frac{d}{u} = 1$, the value of β that minimizes the energy consumption of an optimal scheme is $\beta = \min\left\{\sqrt{\frac{PB}{u\delta}}, n\right\}$.
Note that if the value of $\sqrt{\frac{PB}{u\delta}}$ is not an integer, it has to be rounded to one of the two closest integer values, such that $E(\beta)$ is minimum.
Intuition: The number of blocks into which the file must be divided depends on the value of δ. If δ is very large, then it is better to divide the file in a small number of blocks, since each block transmission consumes additional energy δ. On the other hand, if δ is small, we can divide the file into a number of blocks such that the energy consumed is reduced due to concurrent transfers. For details, please refer to [31].

IV. DOWNLOAD CAPACITY > UPLOAD CAPACITY

In this section, we consider an energy *homogeneous* system in which $k > 1$. In this scenario we provide lower bounds on the energy consumed by an optimal scheme. Then, we show that the algorithms for $k = 1$ are optimal if $\beta \leq n$ and present a quasi-optimal algorithm for $\beta > n$.
Lower Bound In this setting, the possibility to download more than one block in a slot implies that the minimum number of slots in which a host has to be on can be less than β.

Theorem 6: Let z be an optimal schedule in an energy homogeneous system. Then the energy consumed by z satisfies $E(z) \geq n(\beta + 1) \cdot \Delta$.
Intuition: The derivation of this bound is based on proving that the required number of tree slots is at least n, because there are n clients. For the complete proof, please refer to [31].
(Quasi-)Optimal Distribution Schemes The energy consumption of Algorithms 3 and 1 in an energy homogeneous system with $\beta \leq n$ is exactly $n(\beta + 1)\Delta$ (Theorem 4). Hence, these algorithms describe optimal schemes for this system. However, if $\beta > n$, the algorithm for $k = 1$ (Algorithm 2) is not optimal anymore if $k > 1$. So we present Algorithm 4, that describes a distribution scheme for this case. Note that the scheme uses $k = 2$ only.

Algorithm 4 distributes the file among the clients using ideas from Algorithms 2 and 1. We represent the state of process with a two dimensional array A of size $n \times \beta$ (Fig. 3) with the rows and the columns representing the clients and the blocks, respectively. We set an entry $A_{ij} = 1, i \in \{0, 1, .., n-1\}, j \in \{0, 1, .., \beta - 1\}$ if and only if H_i has received b_j, and 0 otherwise. At the beginning, all the entries are 0 and after the completion of the algorithm they all should be 1. Furthermore, imagine the array A divided in $\lfloor \frac{\beta}{n} \rfloor - 1$ square subarrays of size $n \times n$ and one rectangular subarray of size $n \times (n+b)$. (Note that this is just a conceptual division to understand Algorithm 4 in terms of Algorithms 2 and 1.)

After the first loop, the diagonal of the first square subarray is set to 1, i.e., $A_{ii} = 1, \forall i \in \{0, ..., n-1\}$. Additionally, after the second loop, the top left corner position (see Fig. 3) of each subarray has also been set to 1, i.e., $A_{0j} = 1, \forall j \in \{0, n, 2n, .., (\lfloor \frac{\beta}{n} \rfloor - 1)n\}$. In each iteration of the for loop at Line 10, the elements of one of the subarrays of $n \times n$ are set to 1 by serving in the same fashion as in Algorithm 1, while the server completes serving the diagonal of the next square/rectangular subarray. When Line 17 is reached, all the elements of all the square subarrays are marked as 1. The remaining blocks are served using Lines

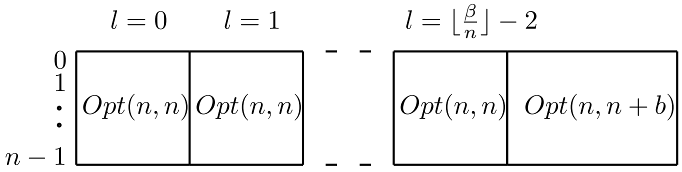


Figure 3. A representation of Algorithm 4 to visualize the distribution of blocks using the ideas of Algorithm 1 and 2.

Algorithm 4 Energy saving scheme for case $k = 2$ and $\beta > n$

```
$b = \beta \bmod n$
for $j = 0 : n-1$
    begin slot
    $S \xrightarrow{j} H_j$
    end slot
for $j = 1 : \lfloor \frac{\beta}{n} \rfloor - 1$
    begin slot
    $S \xrightarrow{nj} H_0$
    end slot
for $l = 0 : \lfloor \frac{\beta}{n} \rfloor - 2$
    for $j = 0 : n-2$
        begin slot
        $S \xrightarrow{(l+1)n+j+1} H_{j+1}$
        for $i = 0 : n-1$
            $H_i \xrightarrow{ln+((i+j) \bmod n)} H_{(i-1) \bmod n}$
        end slot
Run Lines 3-9 of $Opt(n, n+b)$ after renaming the block $b_{\beta-(n+b)+j}$
    to $b_j$, $\forall j \in \{0, 1, .., n+b-1\}$
```

3-9 of Algorithm 2, with an appropriate relabeling of the blocks.

We present the bounds achieved in this section in the following theorem. The proof of the second claim can be found in [31].

Theorem 7: In a homogeneous system with $k > 1$,

- If $\beta \leq n$, then Algorithms 3 and 1 describe optimal distribution schemes with energy $E(z) = n(\beta+1) \cdot \Delta$.
- If $\beta > n$, let $b = \beta \bmod n$, then Algorithm 4 describes a distribution scheme with energy

$$E(z) = \left(n(\beta+1) + \left\lfloor \frac{\beta}{n} \right\rfloor + b - 1 \right) \cdot \Delta \qquad (3)$$

While Algorithm 4 does not achieve optimal energy when $\beta > n$, it is quasi-optimal (in addition to asymptotically optimal), since it is off from the lower bound by an additive term of $(\lfloor \beta/n \rfloor + b - 1)\Delta$, which is usually much smaller than the term $n(\beta+1)\Delta$. It is important to note that Algorithm 4 uses $k = 2$. Then, the upper bounds on the minimum energy presented here hold for all values of $k > 1$.

V. PERFORMANCE EVALUATION

In order to assess the performance of our scheme, we have run an extensive simulation study with two objectives. First, to evaluate quantitatively the results of our analysis. Second, to understand the impact on the performance of our schemes of some effects not considered in our analysis, but typical of real scenarios. Due to space restrictions, some of the simulations results are presented in [31].

Experimental Setup In our experiments we have considered two different **scenarios,** corresponding to two different application contexts for the file distribution problem.

First, we consider a *homogeneous scenario* in which all the hosts participating in the file distribution process have the same configuration. Specifically, we have considered the following values for the relevant input parameters in our experiments: nominal power $P = 80$ W, $\delta = 1$ Joule, and upload and download capacity $u = d = 10$ Mbps. Finally, unless otherwise stated, we consider a scenario with one server and 200 hosts. This homogeneous scenario models a corporate network in which both the network infrastructure and the whole set of devices belong to the same company/organization, and are centrally managed.

Then, we consider a *heterogeneous scenario* that captures the case in which hosts are typical Internet nodes (including home users), and it is therefore characterized by a significant variability across hosts in both the energy consumption profile and the observed network performance (i.e. different access speed and congestion conditions). In this setting we assume $u_i = d_i, \forall i \in \mathcal{I}$. In order to simplify our study, in our experiments we consider separately the effect of heterogeneity in power consumption and the effect of varying network conditions.

The **file distribution schemes** that we have considered in the performance evaluation are three. The first one is the file distribution scheme detailed in Section III, called *Opt* here. The second is *Parallel*, which is a scheme in which all users download the same file at the same time from the same server in parallel. This is one of the most common architectures for file distribution. Finally, in the *Serial* scheme the server uploads in sequence the complete file to the hosts involved in the file distribution process. That is, the server uploads the complete file to the first host. Once it finishes, it uploads the file to the second host, and so on.

For our experiments we considered two different **energy models.** In a first one, the hosts only have two power states: an *OFF* state, in which they do not consume power, and an *ON* state, in which they consume the full nominal power, equal to 80 W (typical nominal power consumption for notebooks and desktop PCs lies in the range 60 to 80 W [32]). Unless otherwise stated, this is the default energy model for our experiments.

In order to understand the impact of load proportional energy consumption in our schemes, we consider a model that fits most of the current network devices [32], in which the energy consumed has some dependency on the CPU utilization and network activity. This energy model is characterized by four states. Besides the OFF state, the other states are: the *IDLE* state, in which the device is active but not performing any task, and consuming 80% of the nominal power; the *TX-or-RX* state, in which the device is active and

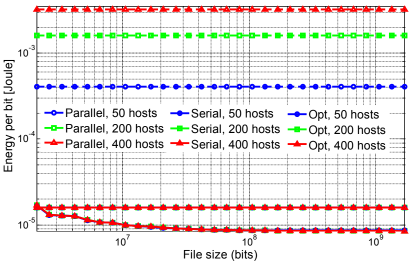


Figure 4. Energy per bit consumed by our algorithm in function of file size, compared with the serial and the parallel scheme. Block size: 256kB.

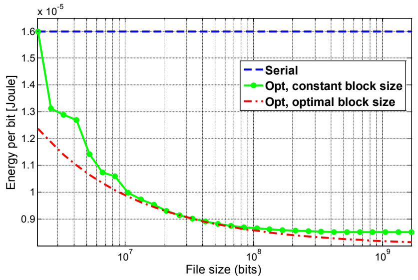


Figure 5. Impact of the choice of number of blocks on the energy per bit consumed by our algorithm, in function of file size, with 200 hosts.

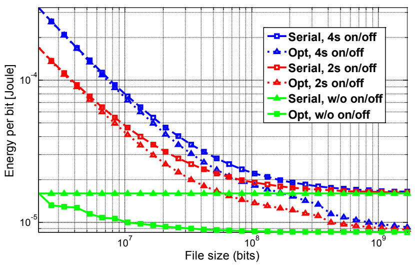


Figure 6. Impact of on/off energy cost on the energy per bit consumed by our algorithm, in function of file size with 200 hosts. Block size: 256kB.

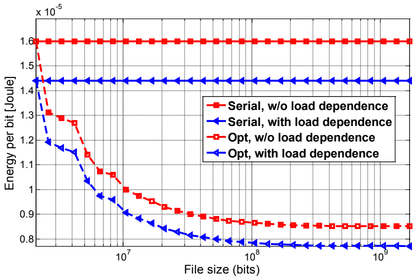


Figure 7. Impact of the energy model on the energy per bit consumed by our algorithm, in function of file size, with 200 hosts. Block size: 256kB.

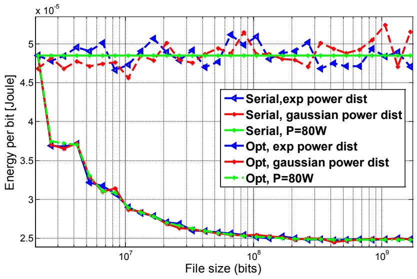


Figure 8. Impact of heterogeneity in nominal power on the energy per bit consumed by our algorithm, in function of file size, with 200 hosts. 95% confidence interval. Block size: 256kB.

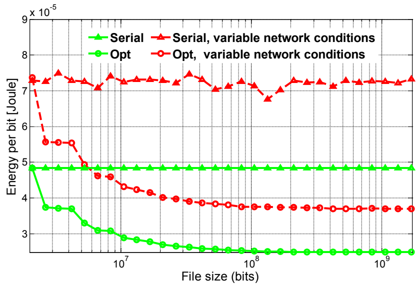


Figure 9. Impact of variable network conditions on the energy per bit consumed by our algorithm, in function of file size, with 200 hosts. 95% confidence interval. Block size: 256kB.

either transmitting or receiving, and consuming 90% of the nominal power; the *TX-and-RX* state, in which the device is active and both transmitting and receiving, and consuming its full nominal power. We considered this model to analyze the impact of load proportionality on the overall energy consumption of the schemes considered in our experiments. Due to space restrictions, the evaluation of the impact of ON/OFF energy cost and the dependency on the load can be found in [31]

In the heterogeneous scenario we analyze the effect of having devices with heterogeneous power consumption profiles. For this purpose we use the previously described two-state model, but we assume that for each host its nominal power consumption is drawn from two different distribution: (i) a Gaussian distribution with an average of 80 W and a standard deviation of 20 W, and (ii) an exponential distribution, with an average of 80 W.

The **goodness metric** we have used in order to compare the energy consumption of different file distribution schemes is *energy per bit*, computed as the ratio of the total amount of energy consumed by the distribution process, divided by the sum of the sizes of all the files delivered in the scheme.

Homogeneous Scenario In order to validate the analysis, in Fig. 4 we have plotted the energy per bit consumed by the file distribution process as function of the size of the file, for the three different file distribution schemes considered. As we can see, our schemes perform consistently better than both serial and parallel schemes. In particular, by maximizing the amount of time in which hosts serve while being served, our schemes tend towards reducing by half the total energy cost of serving a block with respect to the serial scheme. This performance improvement with respect to the serial scheme is due to the use of p2p-like distribution, and indeed it decreases as the file size (and the number of blocks into which it is split) decreases.

Moreover, we can also observe how the parallel scheme performs consistently worse than any other scheme, consuming up to two orders of magnitude more than the serial scheme. Since the utilization of this parallel scheme is widespread in the current Internet, our observations confirm the great potential of distributed schemes for saving energy.

Fig. 4 also depicts the performance of our *Opt* algorithm for different number of hosts (50, 200, and 400). We observe that the energy per bit consumed by our algorithm as well as by the serial scheme are not affected by the number of hosts in the scheme. Hence for the rest of the section we will present results exclusively for a setting with 200 hosts.

The impact of the total number of blocks on the energy consumed by our *Opt* scheme can be seen in Fig. 5, where we plotted the energy per bit consumed with *Opt* for variable file sizes, and for a total of 200 hosts. The green curve corresponds to the case in which a fixed block size, equal to 256 kB, is used, while the lower red one is obtained by using an optimal block size, according to Theorem 5. We see how the use of an optimal block size leads to an increment in energy savings mainly for small file sizes. The reason is that, for small file sizes, a fixed block size leads to a small number of blocks. Consequently, there is less potential parallelism in the (p2p-like) mechanisms, which limits the efficiency of the distribution process.

Heterogeneous Scenarios We consider two separated heterogeneous scenarios. On the one hand, we study the case in which different hosts present different power consumption profiles. On the other hand, we address the scenario in which each host observes different network conditions (i.e., different access speed and congestion level). We note that confidence intervals have been calculated for each curve presented (but not shown for clarity), being in all cases lower than 5%.

In Section III we have proved analytically that our *Opt* algorithm minimizes the overall power consumption of the file distribution process, *even in a heterogeneous scenario in which each host presents a different energy consumption* (as long as all the nodes have the same upload and download rate). To validate this statement, in this subsection we have run experiments in which the nominal power consumed by the hosts varies according to either a Gaussian or an exponential distribution as defined above. Then, the energy consumption has been compared with a homogeneous scenario. The results, presented in Fig. 8, validate our analysis, since the three curves for the *Opt* scheme overlap perfectly. We also observe that heterogeneous power consumption has some minor impact in the case of the serial scheme.

In the results presented we have considered (i) similar upload/download access speed for all host and (ii) no network congestion. We relax now these assumptions, and consider a heterogeneous scenario where hosts have different access speeds and observe different network state (e.g., congestion). This scenario accurately models a content distribution process in the Internet. In particular, in the simulations we model the different nominal access speed of hosts using an exponential distribution, based on realistic speed values provided in [33]. Additionally, in order to model the variation in link speed over time due to network conditions (i.e., congestion) we multiply the nominal access speed by a positive factor taken from a Gaussian distribution with average 1 and standard deviation 0.07. Fig. 9 presents the results for these heterogeneous network conditions, for both our *Opt* scheme and the serial scheme, and compares them with the homogeneous case. The results show that both schemes suffer from an increment in the power consumption, with respect to the homogeneous case. However, the relative difference between the *Opt* and serial schemes increases. This suggests that even in heterogeneous network conditions the proposed algorithm outperforms any centralized scheme. Moreover, we observe that the energy per bit consumed is constant for both *Opt* and serial schemes when considering heterogeneous network conditions. This occurs because none of the considered schemes takes into account host upload/download capacity in determining the schedule for file distribution.

VI. CONCLUSIONS

This paper presents one of the first dives into a novel and relevant field that has received little attention so far: energy-efficiency in file distribution processes. We present a theoretical framework that constitutes the analytical basis for the design of energy-efficient file distribution protocols. Specifically, this framework reveals two important observations: (i) the general problem of minimizing the energy consumption in a file distribution process is NP-hard and (ii) in all the studied scenarios there exists a collaborative distributed algorithm that reduces the energy consumption with respect to popular centralized approaches. This suggests that in those file distribution processes in which reducing the energy consumption is of significant importance (e.g., software updates over night in a corporative network) a distributed algorithm should be implemented.

ACKNOWLEDGEMENTS

This research was supported in part by the European Union FP7 grant 257740 (TREND), the Comunidad de Madrid grants S-2009/TIC-1692 and S-2009/TIC-1468, the Spanish MICINN grant TEC2011-29688-C02, and the National Natural Science Foundation of China grant 61020106002.

REFERENCES

[1] *ICT and CO2 emissions*. [Online]. Available: http://www.parliament.uk/documents/post/postpn319.pdf

[2] L. Chiaraviglio, M. Mellia, and F. Neri, “Minimizing isp network energy cost: Formulation and solutions,” *IEEE/ACM Transactions on Networking,*, 2011.

[3] E. Goma, M. Canini, A. Lopez Toledo, N. Laoutaris, D. Kostić, P. Rodriguez, R. Stanojević, and P. Yagüe Valentin, “Insomnia in the access: or how to curb access network related energy consumption,” in *ACM SIGCOMM*, 2011.

[4] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, “Elastictree: saving energy in data center networks,” in *NSDI*, 2010.

[5] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, “Greening the internet with nano data centers,” in *ACM CoNEXT*, 2009.

[6] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, “Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures,” *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 2, pp. 223–244, 2011.

[7] J. Restrepo, C. Gruber, and C. Machuca, “Energy profile aware routing,” in *Communications Workshops, IEEE ICC 2009.*, 2009, pp. 1–5.

[8] M. Andrews, A. Fernández Anta, L. Zhang, and W. Zhao, “Routing for energy minimization in the speed scaling model,” in *Transactions of Networking, Accepted for publication in 2011, DOI: 10.1109/TNET.2011.2159864*, 2010.

[9] M. Gupta and S. Singh, “Greening of the internet,” in *SIGCOMM*, 2003.

[10] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta, “Somniloquy: augmenting network interfaces to reduce pc energy usage,” in *NSDI*, 2009, pp. 365–380.

[11] M. Pickavet, W. Vereecken, S. Demeyer, P. Audenaert, B. Vermeulen, C. Develder, D. Colle, B. Dhoedt, and P. Demeester, “Worldwide energy needs for ict: The rise of power-aware networking,” in *ANTS*, 2008.

[12] C. Gunaratne, K. Christensen, and B. Nordman, “Managing energy consumption costs in desktop pcs and lan switches with proxying, split tcp connections, and scaling of link speed,” *Int. J. Netw. Manag.*, vol. 15, pp. 297–310, September 2005.

[13] C. Gkantsidis, T. Karagiannis, and M. VojnoviC, “Planet scale software updates,” in *ACM SIGCOMM*, 2006.

[14] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, “Internet inter-domain traffic,” in *ACM SIGCOMM*, 2010.

[15] “Sandvine fall 2011 global internet phenomena report,” http://www.sandvine.com/news/global_broadband_trends.asp.

[16] R. Kumar and K. Ross, “Peer-assisted file distribution: The minimum distribution time,” in *IEEE Workshop on Hot Topics in Web Systems and Technologies (HOTWEB 06)*, 2006, pp. 1–11.

[17] M. Lingjun, P. Tsang, and K. Lui, “Improving file distribution performance by grouping in peer-to-peer networks,” *IEEE Transactions on Network and Service Management*, vol. 6, no. 3, pp. 149–162, 2009.

[18] T. Langner, C. Schindelhauer, and A. Souza, “Optimal file-distribution in heterogeneous and asymmetric storage networks,” *SOFSEM 2011: Theory and Practice of Computer Science*, pp. 368–381, 2011.

[19] M. Mehyar, W. Gu, S. Low, M. Effros, and T. Ho, “Optimal strategies for efficient peer-to-peer file sharing,” in *Acoustics, Speech and Signal Processing, ICASSP*, vol. 4, 2007.

[20] S. Sanghavi, B. Hajek, and L. Massoulie, “Gossiping with multiple messages,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4640–4654, 2007.

[21] L. L. A. G. Matthew Ezovski, Ao Tang, “Minimizing average finish time in p2p networks,” in *IEEE Infocom*, 2009.

[22] K.-S. L. Pui-Sze Tsang, Xiang Meng, “A novel grouping strategy for reducing average distribution time in p2p file sharing,” in *IEEE ICC*, 2010.

[23] J. Mundinger, R. Weber, and G. Weiss, “Optimal scheduling of peer-to-peer file dissemination,” *Journal of Scheduling*, vol. 11, no. 2, pp. 105–120, 2008.

[24] U. Lee, I. Rimac, D. Kilper, and V. Hilt, “Toward energy-efficient content dissemination,” *Network, IEEE*, vol. 25, no. 2, pp. 14–19, 2011.

[25] A. Feldmann, A. Gladisch, M. Kind, C. Lange, G. Smaragdakis, and F. Westphal, “Energy trade-offs among content delivery architectures,” in *IEEE Telecommunications Internet and Media Techno Economics (CTTE),*, 2010, pp. 1–6.

[26] J. Blackburn and K. Christensen, “A simulation study of a new green bittorrent,” in *Communications Workshops, ICC*, 2009, pp. 1–6.

[27] A. P. Giuseppe Anastasi, Ilaria Giannetti, “A bittorrent proxy for green internet file sharing: Design and experimental evaluation,” *Computer Communications*, vol. 33, no. 7, pp. 794–802, 2010.

[28] A. Sucevic, L. Andrew, and T. Nguyen, “Powering down for energy efficient peer-to-peer file distribution,” 2011.

[29] Intel, “Enhanced intel speedstep technology for the intel pentium m processor,” Intel White Paper 301170-001, 2004.

[30] “”in windows 7 use sleep to resume the os in 2 seconds”,” http://news.softpedia.com/news/In-Windows-7-Use-Sleep-to-Resume-the-OS-in-2-Seconds-101290.shtml.

[31] K. Verma, G. Rizzo, A. Fernández Anta, R. Cuevas Rumín, and A. Azcorra, “Greening File Distribution: Centralized or Distributed?” *ArXiv e-prints*, Nov. 2011.

[32] B. Nordman and K. J. Christensen, “Greener pcs for the enterprise,” *IT Professional*, vol. 11, no. 4, pp. 28–37, 2009.

[33] “The real connection speeds for internet users across the world (charts),” http://royal.pingdom.com/2010/11/12/real-connection-speeds-for-internet-users-across-the-world/.