This document is published at:

Verma, Kshitiz, Rizzo, Gianluca, Fernández Anta, Antonio, Cuevas Rumín, Rubén, Azcorra, Arturo, Zaks, Samuel, García-Martínez, Alberto. (2017). Energy-optimal collaborative file distribution in wired networks. *Peer-to-Peer Networking and Applications,* 10(4), pp. pp 925–944.

DOI: https://doi.org/10.1007/s12083-016-0453-4

© Springer Science+Business Media New York 2016

Energy-optimal collaborative file distribution in wired networks

Kshitiz Verma[1,2] · Gianluca Rizzo[3] · Antonio Fernández Anta[4] · Rubén Cuevas Rumín[1] · Arturo Azcorra[1,4] · Shmuel Zaks[5] · Alberto García-Martínez[1]

Abstract The impact of the ICT sector in worldwide power consumption is an increasing concern, motivating the research community to devote an important effort to define novel energy efficient networking solutions. Despite file distribution is responsible for a major portion of the current Internet traffic, little effort has been dedicated to address the issue of its energy efficiency so far. Most of the previous literature focuses on optimizing the download time of file distribution schemes (e.g. centralized server-based or distributed peer-to-peer solutions) while it is yet unclear how to optimize file distribution schemes from the point of view of energy consumed. In this paper, we present a general modelling framework to analyze the energy consumption of file distribution systems. First, we show that the general problem of minimizing energy consumption in file distribution is NP-hard. Then, for restricted versions of the problem, we establish theoretical bounds to minimal energy consumption. Furthermore, we define a set of optimal algorithms for a variety of system settings, which exploit the service capabilities of hosts in a P2P fashion. We show that our schemes are capable of reducing at least 50 % of the energy consumed by traditional (yet largely used) centralized distribution schemes even when considering effects such as network congestion and heterogeneous access speed across nodes.

Keywords P2P · File distribution · Energy efficiency · Algorithms · Performance

✉ Kshitiz Verma
vermasharp@gmail.com

Gianluca Rizzo
Gianluca.Rizzo@hevs.ch

Antonio Fernández Anta
antonio.fernandez@imdea.org

Rubén Cuevas Rumín
rcuevas@it.uc3m.es

Arturo Azcorra
azcorra@it.uc3m.es

Shmuel Zaks
zaks@cs.technion.ac.il

Alberto García-Martínez
alberto@it.uc3m.es

1 Universidad Carlos III de Madrid, Madrid, Spain

2 LNMIIT, Jaipur, India

3 HES SO, Valais, Switzerland

4 Institute IMDEA Networks, Madrid, Spain

5 Technion, Haifa, Israel

1 Introduction

1.1 Background

Recent studies have shown that the Information and Communication Technologies (ICT) sector is contributing significantly to the worldwide energy consumption. For instance, its carbon footprint is nowadays comparable to that of the aviation sector [1]. If new energy mechanisms and solutions are not adopted, the energy consumption of the ICT sector is expected to double in the next decade [38]. Specifically, the design objective of the Internet has primarily been performance with respect to time delay, which means that the devices or services were never designed to take energy into account. However, lately the issue of energy has been raised due to environmental considerations, as well as because of the increasing energy costs, motivating the research community to find solutions to this problem. The proposed approaches to achieve energy efficiency in ICT have mainly focused on designing networks and communication elements so that the power consumed is proportional to the traffic load. In particular, the proposed approaches include the design of new energy-efficient hardware [11], energy efficient routing mechanisms [8, 39], putting devices in sleep mode [5, 21], etc. These approaches address important issues in the core of the network. However, they should be complemented with new techniques to save energy in the end systems (i.e., at the edge of the network) which are responsible for the major portion of the Internet power consumption [20, 38]. To this end, we can, for instance, exploit the existing technology that supports low power modes or switching off the devices whenever it is possible [9, 20]. These techniques can be used to approach energy proportionality in end 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 the overall 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.

In this paper, we narrow down the focus to energy consumption in file-sharing applications/systems. First, as demonstrated by previous studies [35], homes and organizations (i.e., end-hosts) are responsible for 75 % of the overall Internet energy consumption, whereas networking devices (e.g., routers) and data centers are responsible for the other 25 %. Note that existing file distribution services, such as peer-to-peer (P2P) file sharing, one-click-hosting (OCH), software release, etc., represent a major fraction of current Internet traffic, ranging beween 18 and 30 percent [4, 19, 27]. File-sharing applications are run by PCs or laptops that are typically wired devices. The combined effect of the two previous arguments suggests that file-sharing applications are responsible for a significant portion of the overall energy consumption in the Internet. In addition, within the context of corporate/LAN networks, operations such as software updates can be defined also as file distribution processes. Most of previous studies in the area of optimizing file distribution services have mainly focused on minimizing the download time [26, 28, 31, 33]. However, those algorithms designed to minimize the download time are not optimal in terms of energy consumption. Energy consumption in P2P systems is a well studied problem as indicated by numerous references in [32], but this study does not include the file distribution problem.

In this work, we concentrate in a special case of file distribution systems, in which all the hosts that are involved in the file distribution process act in a highly collaborative and coordinated fashion. This scenario corresponds, for instance, with a system that belongs to a single entity, like a company or a university. The connectivity between the hosts may be via a private network or the Internet (possibly accessed via a ADSL connection). A real instance of this set up the authors have found is one in which a company has many screens distributed over a whole city, that during shopping hours show the same commercial video. Every night, the video to be shown the next day is downloaded from a server to all the hosts attached to the screens. These hosts are connected to the Internet via ADSL. As we will show here, the fact that all the hosts are controlled by the same entity allows for tight cooperation among them, saving a significant amount of energy in the daily download.

1.2 Our contribution

Our investigation aims at modeling, analyzing and evaluating the performance of energy-efficient file distribution algorithms in a controlled collaborative environment. Our work provides a realistic characterization of the energy minimization problem in a file distribution process from one server to many hosts that tightly collaborate and coordinate to save energy. We first prove that the general version of the problem, with nodes having arbitrary download and upload capacities, is NP-hard. Thus, we restrict our analysis to scenarios in which all the hosts have equal upload capacity and equal download capacity, which we refer to as *homogeneous scenario*. This makes the problem tractable while still valid for real scenarios. Furthermore, we deal with heterogeneous scenarios using simulation experiments. For the homogeneous versions, we are able to derive theoretical lower bounds on the energy required to complete the file distribution process. We propose file distribution algorithms that are optimal or near-optimal for homogeneous scenarios.

The main intuition behind the proposed file distribution algorithm is to activate node uploading in the same slots as downloading occurs. In this way, we try to reduce the high amount of energy spent when keeping a node turned on just for either downloading or uploading, but not both. While this approach definitely reduces the time required to distribute a file to a set of destinations, compared to a centralized scheme, it is worth to note that the proposed algorithm is not just a time-optimization problem, since different PCs may experience different energy consumption. In this case, the time during which high-consuming PCs are active should be reduced more aggressively than the active time for efficient PCs.

Finally, our empirical evaluation through simulation allows us to validate our analytical results and study the impact of relaxing the homogeneity assumptions imposed in the analysis on the performance of the proposed algorithms. The obtained results support our claim that the proposed algorithms reduce the energy consumption in a file distribution process with respect to any centralized file distribution schemes. In particular, the simulations show that, even in scenarios for which they were not designed (e.g., considering heterogeneous energy consumption or network congestion), our collaborative schemes achieve significant energy savings with respect to largely used centralized file distribution systems. These savings range between 50 % and two order of magnitude, depending on the centralized scheme under consideration.

The rest of the paper is structured as follows. Section 2 revises related work. Section 3 provides NP hardness result and the network and energy model along with definitions and terminology used throughout the paper. Sections 4 and 5 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 6, we present our simulation study and Section 7 concludes the paper.

2 Related work

A large amount of effort has been dedicated to study the completion time in a file distribution process [26, 28, 31, 33]. The minimization of the average finish time in P2P networks is considered in [16, 40, 43]. Mundinger et al. [34] 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.

In parallel with above efforts, energy consumption in P2P and collaborative systems has been the topic of study in various references mentioned in [32] and [14]. The energy models in these works have mainly considered, proxying, sleep-and-wake, task allocation optimization at processor level, message reduction, overlay structure optimization, and location-based techniques to reduce energy consumption. We focus on scheduling of the file distribution mechanism such that the hosts minimize the time they are on for receiving the file. This adds one more model to the above list. On one hand, practical studies [6, 10] have discussed and compared the energy consumed by bittorent protocol along with energy efficient solutions. Evaluation of sleep and wake method in Gnutella network is a matter of study in [15]. However, none of them relies on an analytical basis nor proposes energy-optimal algorithms, as is the case of our paper.

Sucevic et al. [41] and Andrew et al. [7] consider an instance of the problem similar to ours, but under a fluid limit model, in which the file can be split into infinitesimally small blocks. We take a complimentary approach, 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, packet headers, etc). As we show in our paper (see Section 4), the dependence on the blocks size and number of the energy consumption of a distribution scheme is non negligible in any practical scenario.

Another papers considering energy efficient peer to peer file sharing are [22] and bittorent [29]. However, their techniques are restricted to one application of file sharing. The authors of [41] 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 in the paper, for such schemes the total energy consumption is up to twice that of the optimal schemes we propose, depending on the specific settings. Another approach to energy efficiency in P2P networks makes use of hierarchies and sleep modes of the hosts [37, 42]. The aim in this approach is to reduce the number of peers that are active.

Another important and relevant studies pertaining to different energy efficient techniques for content distribution architectures are reviewed in [30]. Energy trade-offs for P2P, data center and content distributions are explored in [17]. The authors of [25] use Network Functions Virtualization for energy efficiency by exploiting the prime-time and non prime-time usage of the infrastructure. [12, 13] proposes power savings in content distribution networks (CDNs) by utilizing dynamic power management of the CDN's cache servers. Energy efficiency in data centers is very much desired and is discussed in [23, 44].

3 System model, problem definition and assumptions

3.1 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 $\mathcal{H} = \{S, H_0, H_1, \ldots, H_{n-1}\}$, and the set of blocks as $\mathcal{B} = \{b_0, b_1, \ldots, b_{\beta-1}\}$. 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.

We assume that time in the file distribution process is slotted. Each block transfer lasts one slot. In general, 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 unit duration. Then, if the process of file distribution starts at time $t = 0$, the first slot spans time $[0, 1)$, the second slot $[1, 2)$, and so on. In each slot, a host is assigned a set of hosts to serve (if any), and the set of blocks it will serve during that slot. Some of the notations used in the work are provided in Table 1.

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 [24], cooling, caching and hard disk activity, network card activity, etc. While in practice δ_i may depend on the size of the block, we assumed all block sizes to be the same, thus δ_i is the same for all hosts.
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 [2]. We usually assume these parameters to be 0.

Table 1 Some of the notation used in this work

Symbol	Definition
n	Total number of clients
H_i	i^{th} client
S	Server, host that has the file initially
$\mathcal{I}$	Set of host indexes
β	Number of blocks into which the file is divided
b_j	j^{th} block
B	Size of the file in bits
s	Size of a block in bits
u	Upload link speed (bits/s)
d	Download link speed (bits/s)
k	Ratio of the download to upload capacity (d/u)
P_i	Power consumed by host i when on (in Watt)
Δ_i	Energy consumed by host i involved in a block transfer in a slot
τ	Any arbitrary time slot
z	A scheme to accomplish the distribution process
$c^z_{j,i}$	Energy to transfer block b_j to host H_i under z
$serv(j, i)$	Index of the host that serves b_j to host H_i
$\mathcal{I}^z_\tau$	Set of active hosts in slot τ under scheme z

3.2 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.

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.

We show that the problem is NP-Complete, even if switching on and off is free and there is no additional energy consumption per block (i.e., $\alpha_i = \delta_i = 0, \forall i \in \mathcal{I}$). 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 4) or near optimally (Theorem 8, Section 5).

Theorem 1 *Assume that time is slotted, that a host can upload and download at the same time slot, and that a 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 $\alpha_i = \delta_i = 0, \forall i \in \mathcal{I}$.*

Proof We show a reduction from the following NP-Complete problem (see [18]):
Partition Problem:
Input: A set of integers $A = \{a_1, a_2, \cdots, a_n\}, 0 < a_i < M$ for every i, $\sum_{j=1}^{n} a_j = 2M$.

Question: Is there a subset $\{a_{j_1}, a_{j_2}, \cdots, a_{j_k}\} \subseteq A$ such that $\sum_{t=1}^{k} a_{j_t} = M$?

We are given an instance I of the Partition Problem, that is, $A = \{a_1, a_2, \cdots, a_n\}$, $0 < a_i < M$ for all i, $\sum_{i=1}^{n} a_i = 2M$. We define an instance $\widehat{I}$ of our problem, as follows. The set of hosts is $\{S, R\} \cup N$, where $N = \{1, 2, \cdots, n\}$. S is the server who initially holds the file of $2M$ blocks of size 1. The upload capacities are $(2n + 1)M$ for S, a_i for every $i \in N$, and 0 for R. The download capacities are 0 for S, $2M$ for every $i \in N$, and M for R. The power consumptions are $E_S = M + 1$, $E_i = a_i$ for every $i \in N$, and $E_R = 4M + 2$. The bound for the total energy is $E = 12M + 5$. We have to show that there is a solution to I iff there is a solution to $\widehat{I}$.

Assume there is a solution to I, that is, a subset $\{a_{j_1}, a_{j_2}, \cdots, a_{j_k}\} \subseteq A$ such that $\sum_{t=1}^{k} a_{j_t} = M$. We describe a solution for $\widehat{I}$. First the source S will send the $2M$ blocks to each of the hosts in N, and M blocks to R, in one time slots (note that its upload capacity is $(2n + 1)M$). This will use $E_S + E_R + \sum_{i=1}^{n} E_i = 5M + 3 + \sum_{i=1}^{n} a_i = 7M + 3$ energy. Hosts $j_1, j_2, \cdots, j_k$, whose total upload capacity is $\sum_{t=1}^{k} a_{j_t} = M$, will then send in one time slot the rest M blocks to R. This will use $E_R + \sum_{t=1}^{k} E_{j_t} = 4M + 2 + \sum_{t=1}^{k} a_{j_t} = 5M + 2$ energy. Thus, the total energy used will be $12M + 5 = E$. We have thus established a solution to $\widehat{I}$, which uses no more than E energy.

Assume there is a solution for $\widehat{I}$, that uses no more than $E = 12M + 5$ energy. As R needs to download $2M$ blocks, and can download at most M blocks in one time slot, it must be active in at least two time slots. If it will be active in more time slots, then the total energy consumed will be at least $3E_R$; this is a contradiction, since $3E_R > E$. Thus R must be active in exactly two time slots, and in each of them it must receive exactly M blocks.

The energy used by R in these two time slots is $2E_R$. Also, there is at least one round in which S uploads, and thus uses E_S energy. Last, for each host in $i \in N$, in the first time slot when it downloads blocks it uses E_i energy and does not upload any block. Thus the total energy used is at least $2E_R + E_S + \sum_{i=1}^{n} E_i = 11M + 5$. Thus, a total of $11M + 5$ is used in which R can download at most M blocks. So at most M energy can be used by hosts who upload the other M blocks to R. This can be done only by hosts in N. But the total energy to be used is at least M (since a host $i \in N$ who uploads at most a_i blocks uses $E_i = a_i$ energy). We conclude that R downloads M blocks from hosts in N whose total energy is M. Thus, if in one of these time slots R uploads the M blocks from the k hosts $\{j_1, j_2, \cdots, j_k\} \subseteq N$, this means that $\sum_{t=1}^{k} a_{j_t} = M$. But then the set of k integers $\{a_{j_1}, a_{j_2}, \cdots, a_{j_k}\}$ is a solution to the instance I of the Partition Problem. □

3.3 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 a host does not upload to more than one other host at the same time. Above assumptions make the problem tractable.

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 (u) results in a uniform slot duration, equal to $\frac{s}{u}$, for all the block transfers. (Recall that s is the block size.) 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.

$$\Delta_i = \frac{P_i s}{u} + \delta_i = \frac{P_i B}{u\beta} + \delta_i. \qquad (1)$$

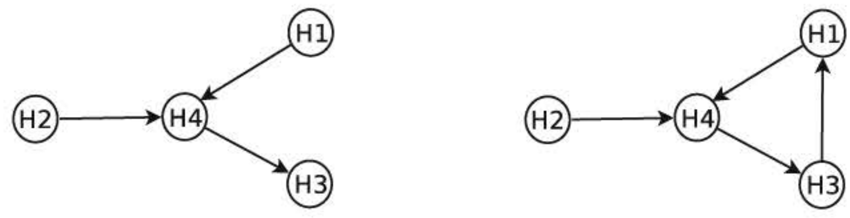
Without loss of generality, we assume that $\Delta_0 \leq \cdots \leq \Delta_{n-1}$.

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.

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.

3.4 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 (i.e., no hosts are powered on without data transfers), there are no slots without active hosts (i.e., data transfer takes place in every slot), 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 as far as energy consumption is concerned by concentrating only



(a) Tree slot (b) Slot with a cycle

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

on normal ones. The time taken to finish the distribution process may increase, though, as two transfer graphs with mutually exclusive hosts cannot exist in a slot because of this restriction. Fortunately, as we will see later, proposed optimal schemes are not affected by this restriction.

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 or a graph with exactly one cycle (Fig. 1). Note also that in a slot with one cycle all hosts that are powered on upload blocks, while in a tree slot there is exactly one host that is powered on but does not upload.

3.5 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. Any transfer graph with the same set of nodes but different edge connectivity are equivalent with respect to the energy consumption. For a better insight on energy consumed by 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,

$$c_{j,i}^z = \mathcal{D}_{j,i}^z \cdot \Delta_i + \mathcal{U}_{j,i}^z \cdot \Delta_{serv(j,i)} \tag{2}$$

where, if b_j is received by H_i in slot τ,

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

$\mathcal{D}_{j,i}^z$ 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.

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*

$$E(z) = \sum_{i=0}^{n-1} \sum_{j=0}^{\beta-1} c_{j,i}^z = \sum_{i=0}^{n-1} \sum_{j=0}^{\beta-1} \left(\Delta_i \cdot \mathcal{D}_{j,i}^z + \Delta_{serv(j,i)} \cdot \mathcal{U}_{j,i}^z \right) \tag{3}$$

Proof We transform the cost of a block as defined in Eq. 2 to the following one. For each host $i \in \mathcal{I}_\tau^z$, define ϕ_i and ψ_i as

$$\phi_i = \begin{cases} \Delta_i & \text{if } \mathcal{S}_{i,\tau}^z \neq \emptyset \\ 0 & \text{Otherwise} \end{cases} \quad \psi_i = \begin{cases} \Delta_i & \text{if } \mathcal{S}_{i,\tau}^z = \emptyset \\ 0 & \text{Otherwise} \end{cases}$$

Note that $\sum_{b_j \in \mathcal{S}_{i,\tau}^z} \mathcal{D}_{j,i}^z = 1$ iff $|\mathcal{S}_{i,\tau}^z| \geq 1$ (i.e., when $\phi_i = \Delta_i$). It is easy to see that $\mathcal{U}_{j,i}^z = 1$ iff $\psi_{serv(j,i)} = \Delta_{serv(j,i)}$, i.e., $S_{serv(j,i),\tau}^z = \emptyset$. Therefore, for a host $i \in \mathcal{I}_\tau^z$, either $\phi_i = \Delta_i$ or $\psi_i = \Delta_i$, never both 0 or both Δ_i. Hence,

$$\sum_{i \in \mathcal{I}_\tau^z} (\phi_i + \psi_i) = \sum_{i \in \mathcal{I}_\tau^z} \Delta_i$$

4 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.

4.1 Lower bound

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

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 to the clients.

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 $i = 0 : n-1$
> $H_i \xrightarrow{(i+j) \bmod n} H_{(i-1) \bmod n}$

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 $E(z) \geq \beta \left(\Delta_S + \sum_{i=0}^{n-1} \Delta_i\right) + \max\{0, n-\beta\} \min\{\Delta_S, \Delta_0\}$.*

Proof The claim to be shown is that if $k = 1$ any scheme z consumes energy

$$E(z) \geq \beta \left(\Delta_S + \sum_{i=0}^{n-1} \Delta_i\right) + \max\{0, n-\beta\} \min\{\Delta_S, \Delta_0\} \quad (4)$$

Before proving the claim, we need some supporting arguments.

Lemma 1 *For every block b_j and every client H_i it holds that $\mathcal{D}^z_{j,i} = 1$.*

Proof Since $d = u$, each host can receive only one block in a time slot. Hence, if block b_j is transferred to client H_i in slot τ, we have $|\mathcal{S}^z_{i,\tau}| = 1$. Then, by definition, $\mathcal{D}^z_{j,i} = 1$. □

Lemma 2 *For every block b_j served by S to client H_i, it holds $\mathcal{U}^z_{j,i} = 1$.*

Proof Let S be serving b_j to H_i in slot τ. Then, $\mathcal{S}^z_{S,\tau}$ is always $\emptyset$, because the server never receives any block from the clients, which means that $\mathcal{U}^z_{j,i} = 1$ for any block b_j served by S. □

Since S has to serve each block of the file at least once, we obtain the following corollary.

Corollary 1 *For at least β block transfers $\mathcal{U}^z_{j,i} = 1$.*

Lemma 3 *If there exists a host H that is receiving its first block in a time slot τ, then there is at least one block b_j in τ such that $\mathcal{U}^z_{j,i} = 1$.*

Proof The number of active hosts in slot τ is $|\mathcal{I}^z_\tau|$. At most $|\mathcal{I}^z_\tau|-1$ blocks can be transferred in τ because host H cannot upload to anyone. Then, since $d = u$, there exists at least one host H' that is on only for uploading. Let b_j be the block served by H'. As it is not downloading any block, $\mathcal{S}^z_{H',\tau} = \emptyset$ and hence $\mathcal{U}^z_{j,i} = 1$. □

Corollary 2 *There are n hosts that receive a block for the first time. Thus, for at least n block transfers $\mathcal{U}^z_{j,i} = 1$.*

We now prove the claim. In order to compute the minimum energy consumption, we need to lower bound Eq. 3. From Lemma 1, it follows that

$$\sum_{i=0}^{n-1} \sum_{j=0}^{\beta-1} \Delta_i \cdot \mathcal{D}^z_{j,i} = \beta \cdot \sum_{i=0}^{n-1} \Delta_i. \quad (5)$$

From Lemma 2 and Corollaries 1 and 2,

$$\sum_{i=0}^{n-1} \sum_{j=0}^{\beta-1} \Delta_{serv(j,i)} \cdot \mathcal{U}^z_{j,i} \geq \beta \cdot \Delta_S + \max\{0, n-\beta\} \cdot \min\{\Delta_S, \Delta_0\}. \quad (6)$$

Adding Eqs. 5 and 6, the claim follows.

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.

```
for slot j = 0 : β − 1
    S →(j) H_j
for slot j = β : n − 1
    H_min →(0) H_{j+1−β}
    for i = 1 : β − 1
        H_{i+j−β} →(i) H_{i+j+1−β}
for slot j = n : n + β − 2
    H_{2n−(j+1)} →(β−1) H_{n+β−(j+2)}
    for i = 0 : β − 2
        H_{(n+i−j) mod n} →(i) H_{(n+i−j−1) mod n}
```

4.2 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 $H \stackrel{j}{\rightarrow} H'$. While the three algorithms could be merged into one, we have chosen to present them separately for clarity.

Theorem 4 *When $d = u$, Algorithms 1, 2 and 3 describe optimal distribution schemes, where*

- *the energy consumed is* $E(z) = \beta\left(\Delta_S + \sum_{i=0}^{n-1}\Delta_i\right) + \max\{0, n-\beta\}\min\{\Delta_S, \Delta_0\}$,
- *each host is on exactly β slots, except $H_{\min}$ that is on* $\max\{\beta, n\}$ *slots (where $H_{\min} \in \{S, H_0\}$ is the host with smallest Δ_i), and*
- *no host is switched on (and off) more than thrice (twice in Algorithms 1 and 2).*
- *energy consumed by host i in Algorithms 1 and 2 is equal to $\beta \cdot \Delta_i$. In Algorithm 3, $H_{\min}$ consumes $n\Delta_{min}$ and everyone else consumes $\beta\Delta_i$.*

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 slot uploads block b_0 to those hosts without any block.

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

4.3 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 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. 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* $\beta = \min\left\{\sqrt{\frac{PB}{u\delta}}, n\right\}$.

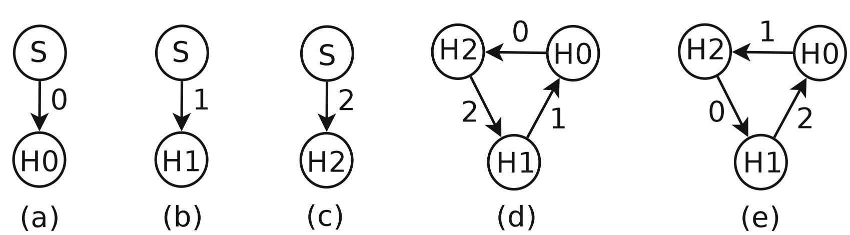


Fig. 2 Example of Algorithm 1, for $n = 3$ and $\beta = 3$. The label on each arrow is the index of the block being served

Proof From Theorems 3 and 4, the energy consumption of an optimal scheme z in an energy homogeneous system is

$$E(z) = (n\beta + \max\{n, \beta\}) \cdot \left(\frac{PB}{u\beta} + \delta\right) \tag{7}$$

To find the optimal value of β, we need to minimize the right hand side of Eq. 7. This can be written as a function of β as

$$E(\beta) = \begin{cases} \frac{PB}{u}(n+1) + \delta(n+1)\beta, & \beta \geq n \quad (8) \\ \frac{nPB}{u}\left(1 + \frac{1}{\beta}\right) + \delta n(\beta+1), & \beta \leq n \quad (9) \end{cases}$$

When this value is larger than n the value $\beta = n$ has to be used. 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. □

5 Download capacity > upload capacity

In this section, we consider an energy *homogeneous* system in which $k > 1$. From Theorem 7, it is evident that the algorithms for $k = 1$ are optimal if $\beta \leq n$. We present a quasi-optimal algorithm for $\beta > n$.

5.1 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. However, we prove the following two important theorems that characterize the behavior of optimal schemes in a homogeneous setting for any $k \geq 1$.

Theorem 6 *A scheme for homogeneous system is optimal if and only if it minimizes the number of tree slots.*

Proof A scheme is optimal $\Rightarrow$ It minimizes the number of tree slots.

Consider a scheme $z \in \mathcal{Z}_k^{n,\beta}$ that has T tree slots. Also assume that z finishes in S slots. Let the cost of slot τ be c_τ^z if n_τ blocks are transfered in it. Note that there can be only two kinds of slots in normal schemes, either a slot with a cycle or a tree slot. If slot τ is a tree slot, then

$$c_\tau^z = (n_\tau + 1) \cdot \Delta \tag{8}$$

If slot τ is slot with a cycle, then

$$c_\tau^z = n_\tau \cdot \Delta \tag{9}$$

So the cost of z, $c(z)$ is given by

$$c(z) = \sum_{\tau=1}^{S} c_\tau^z = \left(\sum_{\tau=1}^{S} n_\tau + T\right) \cdot \Delta = (n\beta + T) \cdot \Delta \tag{10}$$

Since z is optimal, $n\beta$ is the total number of blocks to be transfered. Clearly, $c(z)$ is minimized for $T = T_{min}$ for any k and given n and β.

If a scheme minimizes the number of tree slots $\Rightarrow$ The scheme is optimal.

It is trivial to see that if a scheme z minimizes the number of tree slots then its energy consumption can be given by $c(z) = (n\beta + T_{min}) \cdot \Delta$. No scheme that has a lesser cost can exist because $n\beta$ is the number of blocks that must be transfered and T_{min} is the minimum number of tree slots possible. So z must be optimal. □

Theorem 7 *Let z be any scheme for an energy homogeneous system i.e., all the hosts consume same energy per slot. Also let the energy consumed by z be $E(z)$, then*

$$E(z) \geq n(\beta + 1) \cdot \Delta. \tag{11}$$

Proof It follows directly from Theorem 6, if we can prove that the number of tree slots are at least n, i.e., the number of clients. Consider the slot in which client H_i receives the first block. Hence, H_i cannot be part of any cycle. There cannot be a cycle between any other number of hosts in this slot because each host can upload to a maximum of one host. So if there is a cycle somewhere, block to H_i cannot be uploaded in this slot. There are n clients, so there must be at least n tree slots. □

5.2 (Quasi-)Optimal distribution schemes

Note that the lower bound on energy consumption when $\beta \leq n$ presented in Theorem 7 in is the same as the lower bound presented in Theorem 3 for $k = 1$, when applied to energy homogeneous systems. The energy consumption of Algorithms 1 and 3 in an energy homogeneous system with $\beta \leq n$ is exactly $n(\beta + 1)\Delta$ (Theorem 4). Hence, Algorithms 1 and 3, which were optimal for $\beta \leq n$ in case of $k = 1$ are optimal in this case as well.

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 1 and 2. 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 1 and 2.)

After the first loop, the diagonal of the first square subarray is set to 1, i.e., $A_{ii} = 1, \forall i \in \{0, \ldots, n-1\}$. 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 3-9 of Algorithm 2, with an appropriate relabeling of the blocks.

Theorem 8 *In a homogeneous system with $k > 1$,*

- *If $\beta \leq n$, then Algorithm 1 (when $\beta = n$) and Algorithm 3 (when $\beta < n$) describe optimal distribution schemes with energy $E(z) = n(\beta + 1) \cdot \Delta$ (Theorem 7)*
- *If $\beta > n$, let $b = \beta \bmod n$, then Algorithm 4 describes a distribution scheme with energy*

$$E(z) = \left(n(\beta + 1) + \left\lfloor \frac{\beta}{n} \right\rfloor + b - 1 \right) \cdot \Delta \tag{12}$$

- *Energy consumed by S is $\beta \Delta_S$, host 0 is $(\left\lfloor \frac{\beta}{n} \right\rfloor n + b)\Delta$ and by H_i, $\forall i \in \{1, 2, .., n-1\}$ is $(\lfloor \frac{\beta}{n} \rfloor (n-1) + b + 1) \cdot \Delta$. Thus, this algorithm is unfair to host 0 which consumes $(\lfloor \frac{\beta}{n} \rfloor - 1) \cdot \Delta$ more energy compared to the other hosts. Additionally, no host is switched on (and off) more than thrice.*

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

1: $b = \beta \bmod n$
2: for $j = 0 : n - 1$
3:   ***begin slot***
4:   $S \xrightarrow{j} H_j$
5:   ***end slot***
6: for $j = 1 : \lfloor \frac{\beta}{n} \rfloor - 1$
7:   ***begin slot***
8:   $S \xrightarrow{nj} H_0$
9:   ***end slot***
10: for $l = 0 : \lfloor \frac{\beta}{n} \rfloor - 2$
11:   for $j = 0 : n - 2$
12:     ***begin slot***
13:     $S \xrightarrow{(l+1)n+j+1} H_{j+1}$
14:     for $i = 0 : n - 1$
15:       $H_i \xrightarrow{ln+((i+j) \bmod n)} H_{(i-1) \bmod n}$
16:     ***end slot***
17: Run Lines 3–9 of $Opt(n, n + b)$ after renaming the block $b_{\beta-(n+b)+j}$ to $b_j, \forall j \in \{0, 1, .., n+b-1\}$

Note that for $\beta \leq n$, Algorithms 1 and 3 are still optimal even though $k > 1$. This indicates that increasing k does not always result in energy savings. The fact that $k > 1$ is helpful only when the number of blocks is greater than the number of hosts. The intuition is that if a host receives from, say, $k \geq 2$ hosts, it happens at the cost of at least $k - 1$ hosts who cannot receive, because the upload degree of a transfer graph is limited by the number of nodes. This essentially nullifies the effect of parallel uploads to a host in this scenario where all the hosts have equal power consumption.

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 $k = 2$, which again indicates that having high download to upload capacity ratio does not lead to high energy savings.

The upper bounds on the minimum energy presented here hold for all values of $k > 1$. For details of the proof, refer to the Appendix.

6 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

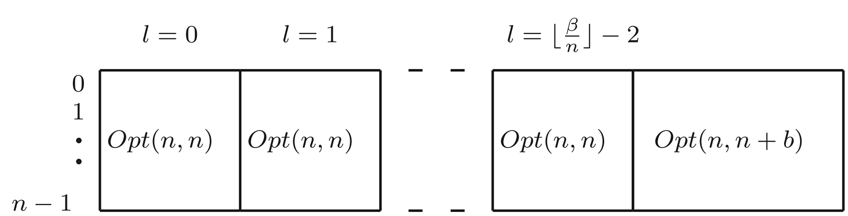


Fig. 3 A representation of Algorithm 4 to visualize the distribution of blocks using Algorithm 1 and 2

of some effects not considered in our analysis, but typical of real scenarios.

6.1 Experimental setup

In our experiments, we assume that each host is connected to every other hosts, i.e., we consider the topology of a complete graph. This is the topology of many application-level overlays made of Internet hosts. We do not consider the intermediate devices like switches or routers in our evaluation, since we assume that they cannot be turned off at will.

At the host level, 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 4, 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.

6.2 Energy model

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 [36]). 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 [36], 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 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.

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.

6.3 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.

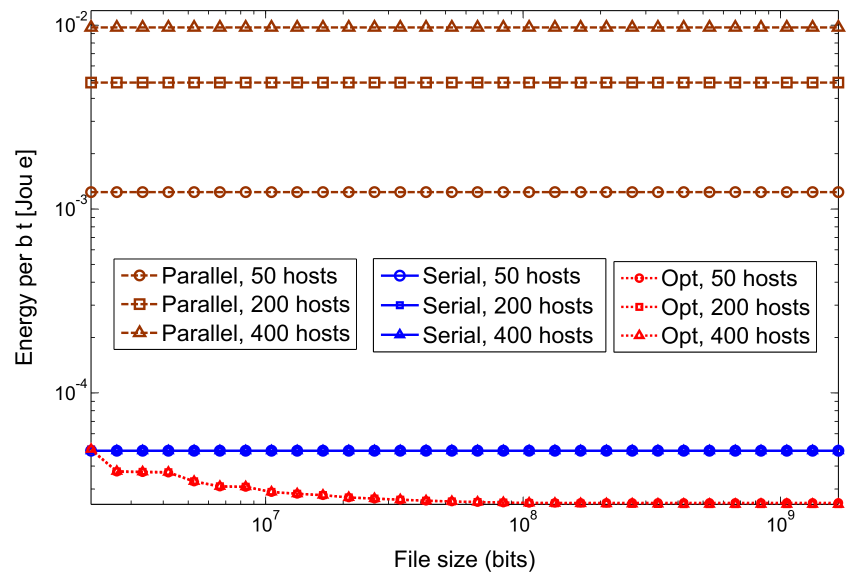


Fig. 4 Energy per bit consumed by our algorithm in function of file size, compared with the serial and the parallel scheme. Block size: 256kB

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.

Figure 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. Note that the curves overlap irrespective of the the number of hosts. 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 (Fig. 4).

ON/OFF Energy Costs As seen in previous sections, our optimal algorithms develop in rounds. Typically, not every host is on in every round (i.e., some go on and off more than once during the file distribution process). In a realistic scenario, a host takes some time to both go off (or into a very low power mode), and to get back to active mode. Usually, this on/off time is in the order of a few seconds [2]. The additional amount of energy consumed while switching between these power states (that we call here "on/off costs") has potentially an important impact on the energy

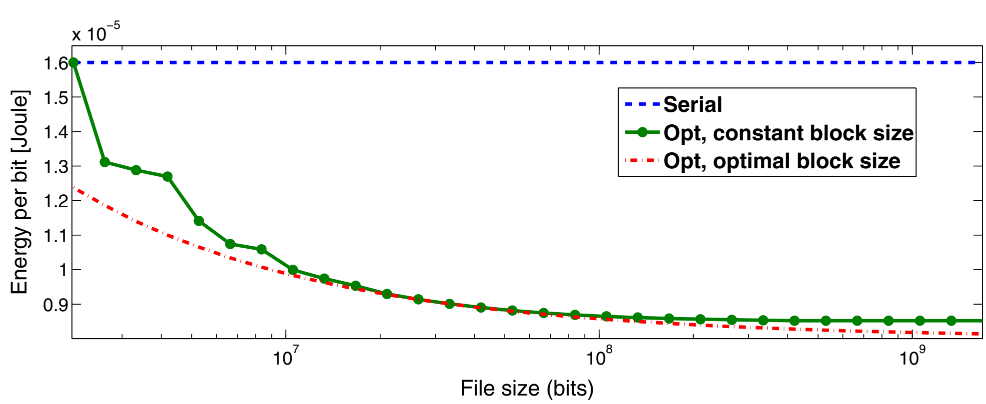


Fig. 5 Impact of the choice of number of blocks on the energy per bit consumed by our algorithm, in function of file size, with 200 hosts

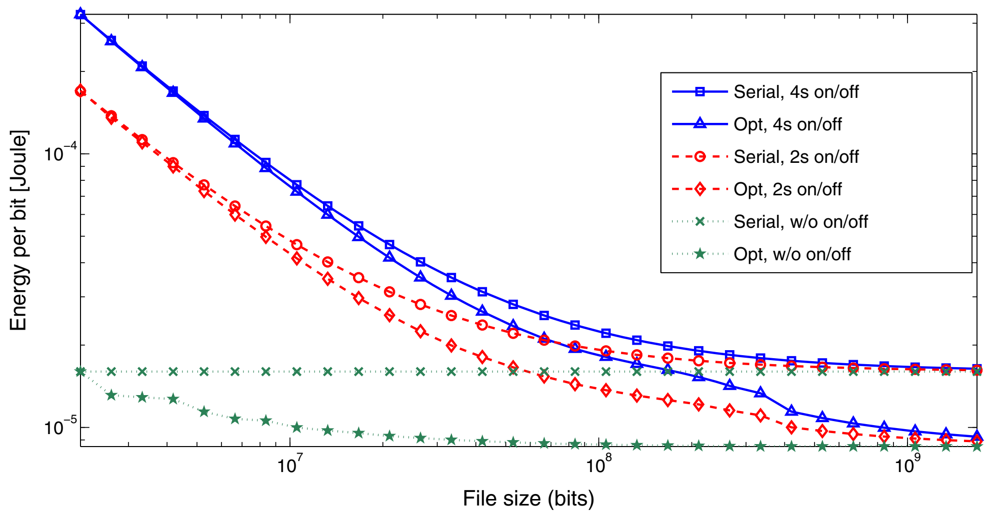


Fig. 6 Impact of on/off energy cost on the energy per bit consumed by our algorithm, in function of file size with 200 hosts. Block size: 256kB

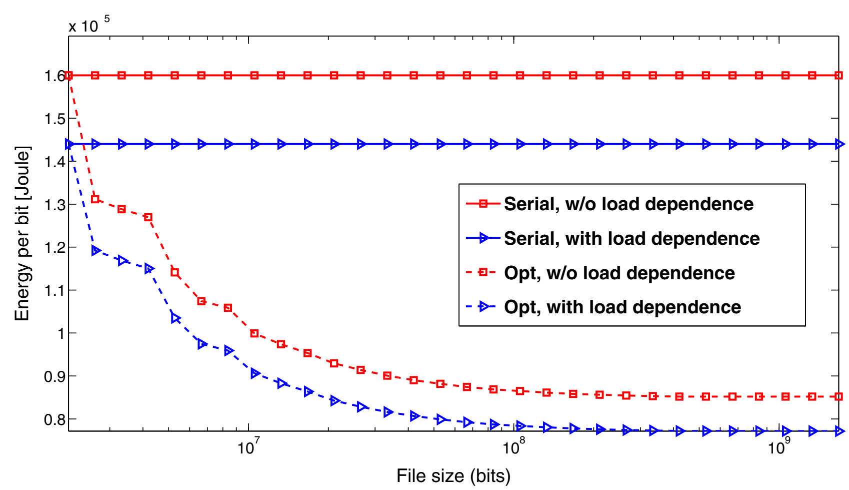
performance of a scheme, penalizing specifically those schemes in which host activity is more "discontinuous" over time.

In order to mitigate the negative impact of on/off costs, in our simulations we implement the following mechanism. When a host A has finished its activity (i.e. uploading and/or downloading a block) in a slot t_1, and has no activity until slot t_2, it computes the energy cost of staying on ($cost_{\text{on}}$) until the slot t_2 and the cost of going off during the rest of slot t_1 and switching on at the beginning of slot t_2 ($cost_{\text{off/on}}$). Hence, if $cost_{\text{on}} \leq cost_{\text{off/on}}$, A decides to stay on. Otherwise, it goes off for its non-active period between slots t_1 and t_2.

Figure 6 presents the energy consumed by our scheme in comparison to the serial scheme considering a switch on/off time equal to 2 and 4s. As expected, the on/off costs increase the energy per bit consumed by all schemes. This increment is more pronounced for small file sizes, where we see that on/off costs make the performance of our scheme closer (but still better) to the serial scheme. Conversely, for medium/large file sizes, the contribution of on/off costs to the total energy consumed by a scheme becomes marginal, and the performance of both the optimal scheme and the serial approaches the one in the case without on/off costs. Note the widening of the gap between the serial scheme and our scheme for file sizes around 50MB is due to the different behavior that our scheme has for the case $n < \beta$ and for the other case.

Load dependency In this set of experiments, we have analyzed the impact of the four-states energy model described in Section 6.2, which implies some degree of energy proportionality of the host devices. The research community is putting a lot of effort in energy proportionality. Hence, in the future it is expected that network devices will consume energy proportionally to the supported load. Figure 7 shows that with the four-states energy model the percentage decrease in the energy per bit consumed by our *Opt* scheme and by the serial one is the same. This suggests that even with load proportional hardware our scheme

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



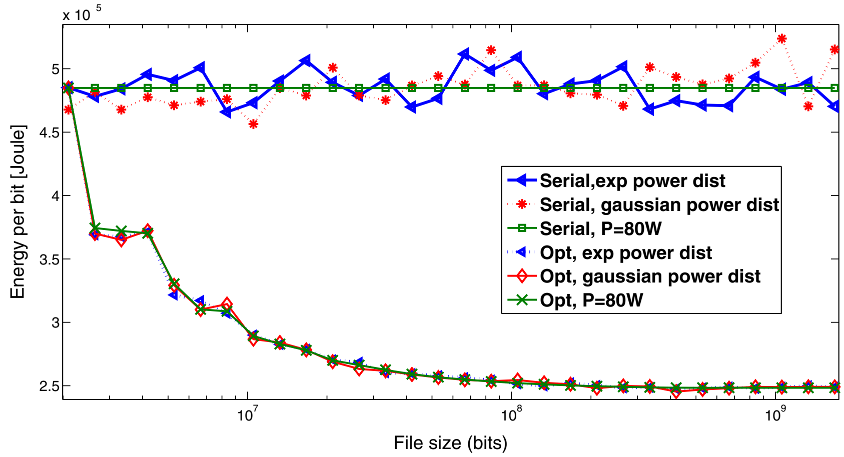


Fig. 8 Impact of heterogeneity in nominal power on the energy per bit consumed by our algorithm, in function of file size, with 200 hosts. 95% confidence interval. Block size: 256kB

enables significant energy savings with respect to the serial one.

6.4 Heterogeneous scenarios

We consider two separated heterogeneous scenarios. On 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 4 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 [3]. 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.

Figure 9 presents the results for these heterogeneous network conditions, for both our *Opt* scheme and the serial

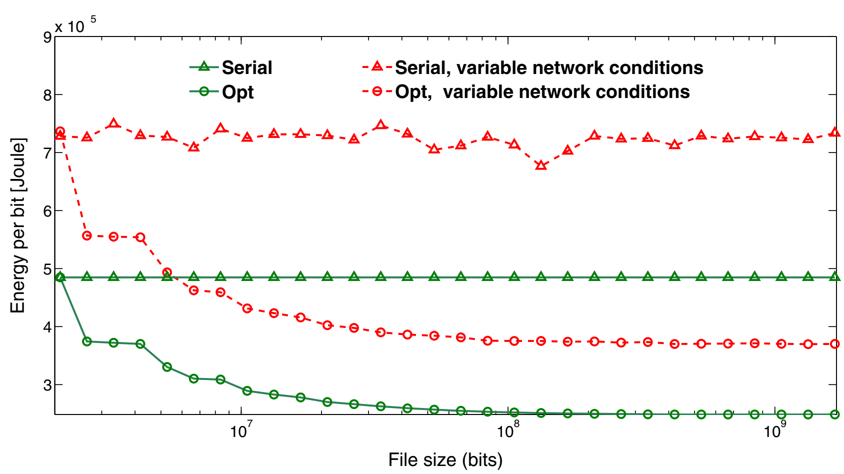


Fig. 9 Impact of variable network conditions on the energy per bit consumed by our algorithm, in function of file size, with 200 hosts. 95 % confidence interval. Block size: 256kB

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.

7 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 distributed collaborative scheme 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 solution should be implemented.

Acknowledgments Supported in part by Ministerio de Economia y Competitividad grant TEC2014- 55713-R, the DRONEXT project (TEC2014-58964-C2-1-R), Regional Government of Madrid (CM) grant Cloud4BigData (S2013/ICE-2894, co- funded by FSE & FEDER), and BRADE Project (P2013/ICE-2958), NSF of China grant 61520106005, and European Commission H2020 grants ReCred and NOTRE.

Appendix A: Proofs of correctness and optimality for $k = 1$

For the correctness and optimality proofs of a scheme z (described by an algorithm), we define the state $\sigma^z_{i,\tau}$ of a host $i \in \mathcal{I}$ at the end of slot τ as the set of blocks held by that time at the host. Thus, to start with, initially for S we have, $\sigma^z_{S,0} = \mathcal{B}$, and, for each client $i \in \{0, \ldots, n-1\}$, $\sigma^z_{i,0} = \emptyset$. If z is correct, after the makespan of z (τ^z_f slots) the state of every client $i \in \{0, \ldots, n-1\}$ must be $\sigma^z_{i,\tau^z_f} = \mathcal{B}$. We omit z and τ when clear from the context.

A.1 Algorithm 2

Let us denote the scheme described by Algorithm 2 as z_2. This scheme has the following properties.

Observation 1 *After the for loop at Lines 1-2,*
(i) the state of client i is $\sigma_i = \{b_i\}, \forall i \in \{0, .., n-1\}$, and
(ii) all hosts, including the server, have been switched on once and switched off once, except host H_{n-1}, which was only switched on once.

Lemma 4 *After the q^{th} iteration of the loop at Lines 3-6, for $q \in \{0, 1, .., \beta - n\}$, each host $H_i, i \in \{0, \ldots, n-1\}$, has state*

$$\sigma_i = \bigcup_{p=0}^{q} \{b_{(i+p)}\} \tag{13}$$

Proof We use induction on q to prove the lemma. The base case ($q = 0$) follows from Observation 1.(i).

Induction step: Assume the hypothesis to be true for the $(q-1)^{\text{th}}$ iteration. Client $H_i, i \in \{0, ..., n-2\}$ receives block $b_{(i+q)}$ in the q^{th} iteration, while client H_{n-1} receives block $b_{(q+n-1)}$ from the server. Thus, $\forall i \in \{0, ..., n-1\}$, the state of client H_i after the q^{th} iteration is

$$\sigma_i = \bigcup_{p=0}^{q-1} \{b_{(i+p)}\} \cup \{b_{(i+q)}\} = \bigcup_{p=0}^{q} \{b_{(i+p)}\}$$

□

Lemma 5 *After the q'^{th} iteration of the loop at Lines 7-9, for $q' \in \{0, 1, .., n-1\}$, each host $H_i, i \in \{0, 1, .., n-1\}$, has state*

$$\sigma_i = \bigcup_{p=0}^{q'+\beta-n} \{b_{(i+p) \bmod \beta}\} \tag{14}$$

Proof We use induction on q' to prove the claim. The base case ($q' = 0$) follows from Lemma 4 with $q = \beta - n$. Let the claim (induction hypothesis) be true for the $(q'-1)^{\text{th}}$ iteration. In the q'^{th} iteration, the value of j is $j = q' + \beta - 1$. Hence, H_i receives block $b_{(i+q'+\beta-n)}$. Thus, the state of client H_i after the q'^{th} iteration is

$$\begin{aligned}\sigma_i &= \bigcup_{p=0}^{q'-1+\beta-n} \{b_{(i+p) \bmod \beta}\} \cup \{b_{(i+q'+\beta-n) \bmod \beta}\} \\ &= \bigcup_{p=0}^{q'+\beta-n} \{b_{(i+p) \bmod \beta}\}\end{aligned} \tag{15}$$

Lemma 6 *During the execution of Algorithm 2 each host $H_i, i \in \{0, ..., n-1\}$ serves a block that it has already downloaded.*

Proof Let us consider the loops at Lines 3-6 and Lines 7-9 in sequence. In the q^{th} iteration of these loops, host H_i serves block $b_{(i+q-1) \bmod \beta}$. From the previous lemmas, after the $(q-1)^{\text{th}}$ iteration of these loops, host H_i has state

$$\sigma_i = \bigcup_{p=0}^{q-1} b_{(i+p) \bmod \beta}$$

which includes $b_{(i+q-1) \bmod \beta}$. Hence the claim follows. □

Theorem 9 *After the termination of Algorithm 2 each host H_i, $i \in \{0, ..., n-1\}$, has received all the blocks $b_j \in \mathcal{B}$ with optimal energy $E(z_2) = \beta(\Delta_S + \sum_{i=0}^{n-1} \Delta_i)$. Additionally, host i consumes exactly $\beta\Delta_i$ energy, and no host has been switched on (and off) more than twice.*

Proof It follows from Lemma 5 that each host has received all the blocks at the end of the loop at Lines 7-9. Then, the scheme is correct since each host serves a block that it has already downloaded (Lemma 6). Each host (including the server) is active exactly β slots. Then, the total energy consumed is $E(z_2) = \beta(\Delta_S + \sum_{i=0}^{n-1} \Delta_i)$, which is optimal since it matches the lower bound. Each host is on for exactly β slots. Hence, the total energy consumed by host i is $\beta\Delta_i$.

It follows from Lemma 4 that all the hosts, including the server, have been on during the execution of the loop at Lines 3-6. Similarly, Lemma 5 means that all the hosts but the server have been on during the execution of the loop at Lines 7-9. This, together with Observation 1.(ii), implies that the server and host H_{n-1} have been switched on (and off) once, whereas the rest of the hosts were switched on/off twice. □

A. 2 Algorithm 3

For the correctness and optimality proofs of Algorithm 3 we define the state $\zeta^z_{r,\tau}$ of a block b_r at the end of τ as the set of clients $H_i, i \in \{0, ..., n-1\}$, who have received b_r. Thus, to start with, $\forall r \in \{0, ..., \beta-1\}$, initially the state of block b_r is $\zeta^z_{r,0} = \emptyset$. After the makespan τ^z_f of scheme z, the state should be, $\forall r \in \{0, ..., \beta-1\}$, $\zeta^z_{r,\tau^z_f} = \bigcup_{i=0}^{n-1}\{H_i\}$

Let us denote the scheme described by Algorithm 3 as z_3. This scheme has the following properties.

Observation 2 *After the for loop at Lines 1-2, $\forall r \in \{0, 1, .., \beta-1\}$, the state of block b_r is $\zeta_r = \{H_r\}$.*

Lemma 7 *After the q^{th} iteration of the for loop at Lines 3-6, for $q \in \{0, ..., n-\beta\}$, the state of block b_r is*

$$\zeta_r = \bigcup_{p=0}^{q}\{H_{r+p}\} \tag{16}$$

Proof We prove the claim using induction on q. The base case ($q = 0$) is trivially true by the observation. Assume the statement to be true for the $(q-1)^{\text{th}}$ iteration. In the q^{th} iteration, $q = j + 1 - \beta$. Then, block b_r is served to H_{r+q}. Thus, the state of block b_r after the q^{th} iteration is

$$\zeta_r = \bigcup_{p=0}^{q-1}\{H_{r+p}\} \cup \{H_{r+q}\} = \bigcup_{p=0}^{q}\{H_{r+p}\}$$

□

Lemma 8 *After the q'^{th} iteration of the for loop at Lines 7-10, for $q' \in \{0, 1, .., \beta-1\}$, the state of block b_r is*

$$\zeta_r = \bigcup_{p=0}^{n-\beta}\{H_{r+p}\} \bigcup_{p=0}^{q'}\{H_{(r-p) \bmod n}\} \tag{17}$$

Proof The base case ($q' = 0$) is true from Lemma 7 after the loop at Lines 3-6 completes. In iteration $q' = j + 1 - n$, block $b_{\beta-1}$ is served to $H_{\beta-q'-1}$, hence,

$$\zeta_{\beta-1} = \bigcup_{p=0}^{n-\beta}\{H_{\beta+p-1}\} \bigcup_{p=0}^{q'-1}\{H_{\beta-1-p}\} \cup \{H_{\beta-1-q'}\}$$

and block b_r, $r \in \{0, 1, .., \beta-2\}$, is served to $H_{(r-q') \bmod n}$. Then, the state of block b_r, $r \in \{0, ..., \beta-1\}$, after the q'^{th} iteration is

$$\begin{aligned}\zeta_r &= \bigcup_{p=0}^{n-\beta}\{H_{r+p}\} \bigcup_{p=0}^{q'-1}\{H_{(r-p) \bmod n}\} \cup \{H_{(r-q') \bmod n}\} \\ &= \bigcup_{p=0}^{n-\beta}\{H_{r+p}\} \bigcup_{p=0}^{q'}\{H_{(r-p) \bmod n}\}\end{aligned}$$

Lemma 9 *During the execution of Algorithm 3, each host $H_i, i \in \{0, 1, .., n-1\}$, serves a block that it has already downloaded.*

Proof In the *for* loop at Lines 3-6, during iteration $q = j + 1 - \beta$, $q \in \{1, .., n-\beta\}$, block b_r is served by H_{r+q-1}. It has it because after iteration $q-1$,

$$\zeta_r = \bigcup_{p=0}^{q-1}\{H_{r+p}\},$$

which includes H_{r+q-1}. H_0 always serves b_0, if any, which it has from the above observation.

In the *for* loop at Lines 7-10, during iteration $q' = j + 1 - n, q' \in \{1, .., \beta - 1\}$, block $b_{\beta-1}$ is served by $H_{n-q'}$. It has it because after iteration $q' - 1$,

$$\zeta_{\beta-1} = \bigcup_{p=0}^{n-\beta} \{H_{\beta+p-1}\} \bigcup_{p=0}^{q'-1} \{H_{\beta-1-p}\} \cup \{H_{\beta-1-q'}\}$$

which includes $H_{n-q'}, \forall q' \in \{1, 2, .., \beta - 1\}$.

Block $b_r, r \in \{0, 1, .., \beta - 2\}$ is served by $H_{(r-(q'-1)) \bmod n}$. It has it because after iteration $q' - 1$

$$\zeta_r = \bigcup_{p=0}^{n-\beta} \{H_{r+p}\} \bigcup_{p=0}^{q'-1} \{H_{(r-p) \bmod n}\}$$

which includes $H_{(r-(q'-1)) \bmod n}$. Hence, the claim follows. □

Lemma 10 *During the execution of Algorithm 3, a host is switched on (and off) at most thrice.*

Proof In each of the for loops at Lines 1-2, 3, 6, 7-10, a host is not switched on (resp. off) more than once, since indices i and j only increase in the loop. There are three such for loops, so a host can be switched on/off at most thrice in Algorithm 3. □

Theorem 10 *After the termination of Algorithm 3 each host H_i, $i \in \{0, ..., n-1\}$ has received all the blocks $b_r \in \mathcal{B}$ with optimal energy $E(z_3) = \beta \left(\Delta_S + \sum_{i=0}^{n-1} \Delta_i\right) + (n - \beta) \min\{\Delta_S, \Delta_0\}$. Additionally, host i consumes exactly $\beta\Delta_i$ energy, except $H_{\min}$ that consumes $n\Delta_{min}$ energy, and no host has been switched on (and off) more than thrice.*

Proof It follows from Lemma 8 that each host has received all the blocks. Then, the scheme is correct since each host serves blocks it has already downloaded (Lemma 9).

We need to bound now the energy consumed. Let us denote $\Delta_{\min} = \min\{\Delta_S, \Delta_0\}$. The energy consumed in the loop at Lines 1-2 is easily observed to be

$$E_1 = \beta\Delta_S + \sum_{i=0}^{\beta-1} \Delta_i \tag{18}$$

The energy consumed in the loop at Lines 3-6 is

$$\begin{aligned} E_2 &= \sum_{j=\beta}^{n-1} \left(\Delta_{\min} + \Delta_{j+1-\beta} + \sum_{i=1}^{\beta-1} \Delta_{i+j+1-\beta} \right) \\ &= (n-\beta)\Delta_{\min} + \sum_{j=0}^{n-\beta-1} \sum_{i=0}^{\beta-1} \Delta_{i+j+1} \end{aligned} \tag{19}$$

Finally, the energy consumed in the loop at Lines 7-10 is

$$\begin{aligned} E_3 &= \sum_{j=n}^{n+\beta-2} \left(\Delta_{n+\beta-j-2} + \sum_{i=0}^{\beta-2} \Delta_{(n+i-j-1) \bmod n} \right) \\ &= \sum_{j=0}^{\beta-2} \sum_{i=0}^{\beta-1} \Delta_{(i-j-1) \bmod n} \end{aligned} \tag{20}$$

Adding Eqs. 18, 19 and 20, we get,

$$\begin{aligned} E(z_3) &= E_1 + E_2 + E_3 \\ &= \beta\Delta_S + (n-\beta)\Delta_{\min} + \sum_{i=0}^{\beta-1} \left(\sum_{j=0}^{i} \Delta_j + \sum_{j=i+1}^{i+n-\beta} \Delta_j \right. \\ &\qquad \left. + \sum_{j=i+n-\beta+1}^{n-1} \Delta_j \right) \\ &= \beta \left(\Delta_S + \sum_{j=0}^{n-1} \Delta_j \right) + (n-\beta)\Delta_{\min}, \end{aligned}$$

which is optimal. This bound implies that all hosts are on exactly β slots, and hence consumes $\beta\Delta_i$ energy except $H_{\min}$ that is on for n slots consuming $n\Delta_{\min}$ energy. Hence, Algorithm 3 is unfair to the host with minimum energy consumption. Finally, the bound of number of times a host is switched on/off is proven in Lemma 10. □

Appendix B: Proof of correctness and performance of algorithm 4

The proof of correctness of Algorithm 4 can be divided in essentially four parts. (We use the array abstraction for clarity.) The first claim is that, after the first loop (Lines 2-5), the diagonal of the first subarray has been filled. (I.e., $A_{ii} = 1, \forall i \in \{0, ..., n-1\}$.) This claim follows trivially by inspection. The second claim is that after the second loop (Lines 6-9), the top left corner position of each subarray has also been set to 1. (I.e., $A_{0j} = 1, \forall j \in \{0, n, 2n, .., (\lfloor \frac{\beta}{n} \rfloor - 1)n\}$.) This claim also follows by inspection.

The third claim is that, after the q^{th} iteration of the third loop (Lines 10-16), the whole q^{th} subarray and the diagonal of the $(q+1)^{\text{th}}$ subarray have been set to 1 (and the blocks served by a host were available at the host for being served). This can be shown by induction on q, where the base case is the first claim above. In the induction step, the proof that the whole q^{th} subarray is set to 1 is similar to the proof of Algorithm 1. The proof that the diagonal of the $(q+1)^{\text{th}}$ subarray is set follows from the second claim above and Line 13 of the algorithm.

Finally, the fourth claim is that the process described in Line 17 completes the array. The proof of this claim is very similar to the proof of Algorithm 2.

Let us now compute the energy consumed by the scheme described by the algorithm. The first loop consumes energy $E_1 = 2n\Delta$, since the server is on n slots, while each client only one. The second loop consumes $E_2 = 2(\lfloor\beta/n\rfloor - 1)\Delta$, since in this loop both the server and the client H_0 are on $\lfloor\beta/n\rfloor - 1$ slots. The third loop uses energy

$$E_3 = \Delta \sum_{l=0}^{\lfloor\frac{\beta}{n}\rfloor-2} \sum_{j=0}^{n-2}(n+1) = \Delta(\lfloor\frac{\beta}{n}\rfloor - 1)(n^2-1),$$

since in this loop all the hosts are on $(\lfloor\beta/n\rfloor - 2)(n-1)$ slots. Finally, the energy consumed by the process described in Line 17 is

$$E_4 = \Delta\left(\sum_{j=n}^{n+b-1}(n+1) + \sum_{j=n+b}^{n+b+n-2} n\right) = \Delta(b(n+1) + n(n-1)).$$

In this process no host is on more than $n+b$ slots. Adding up all these terms we compute the total energy as

$$E(z_4) = \Delta\left(n(\beta+1) + \left\lfloor\frac{\beta}{n}\right\rfloor + b - 1\right).$$

Additionally, we bound the energy consumed by hosts as follows.

– The server is on for exactly β slots, consuming $\beta\Delta_S$ energy.
– The client H_0 is on for exactly $(\lfloor\beta/n\rfloor)n + b$ slots, consuming $((\lfloor\beta/n\rfloor)n + b)\Delta$ energy.
– And the rest of clients are on for exactly $\lfloor\frac{\beta}{n}\rfloor(n-1) + b + 1$ slots, consuming $(\lfloor\frac{\beta}{n}\rfloor(n-1) + b + 1)\Delta$ energy.

Thus, H_0 consumes $(\lfloor\frac{\beta}{n}\rfloor - 1)\Delta$ energy more than any other client. To prove that hosts switch on and off at most three times, the proof is analogous to that for previous algorithms. In the execution of Lines 2-9, all hosts switch on and off at most once except H_0, that switches on and off twice. In the rest of the algorithm, all clients are on until they finish downloading, and the server is switched off as soon as it serves all the blocks.

References

1. ICT and CO_2 emissions. http://www.parliament.uk/documents/post/postpn319.pdf
2. 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
3. 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/
4. Sandvine global Internet phenomena report 2h 2012. http://www.sandvine.com
5. Agarwal Y, Hodges S, Chandra R, Scott J, Bahl P, Gupta R (2009) Somniloquy: augmenting network interfaces to reduce pc energy usage. In: NSDI, pp 365–380
6. Anastasi G, Giannetti I, Passarella A (2010) A bittorrent proxy for green Internet file sharing: Design and experimental evaluation. Comput Commun 33(7):794–802
7. Andrew LL, Sucevic A, Nguyen TT (2011) Balancing peer and server energy consumption in large peer-to-peer file distribution systems. In: IEEE Online Conference on Green Communications (GreenCom), pp 76–81
8. Andrews M, Fernández AA, Zhang L, Zhao W (2012) Routing for power minimization in the speed scaling model. IEEE/ACM Trans Networking (TON) 20(1):285–294
9. Arjona Aroca J, Chatzipapas A, Fernández Anta A, Mancuso V (2014) A measurement-based analysis of the energy consumption of data center servers. In: Proceedings of the 5th international conference on Future energy systems, e-Energy 2014, pp 63–74. ACM
10. Blackburn J, Christensen K (2009) A simulation study of a new green bittorrent. In: Communications Workshops, ICC, pp 1–6
11. Bolla R, Bruschi R, Davoli F, Cucchietti F (2011) Energy efficiency in the future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures. IEEE Commun Surv Tutorials 13(2):223–244
12. Bostoen T, Mullender S, Berbers Y (2015) Lofoswitch: An online policy for concerted server and disk power control in content distribution networks. Ad Hoc Netw 25:606–621
13. Bostoen T, Napper J, Mullender S, Berbers Y (2013) Minimizing energy dissipation in content distribution networks using dynamic power management. In: Third IEEE International Conference on Cloud and Green Computing (CGC), pp 203–210
14. Brienza S, Cebeci SE, Masoumzadeh SS, Hlavacs H, Anastasi G et al. (2015) A survey on energy efficiency in P2P systems: File distribution, content streaming, and epidemics. ACM Comput Surv (CSUR) 48(3):36
15. Corigliano S, Trunfio P (2014) Exploiting sleep-and-wake strategies in the gnutella network. In: IEEE International Conference on Collaboration Technologies and Systems (CTS), pp 406–412
16. Ezovski GM, Tang A, Andrew LL (2009) Minimizing average finish time in P2P networks. In: IEEE Infocom
17. Feldmann A, Gladisch A, Kind M, Lange C, Smaragdakis G, Westphal F (2010) Energy trade-offs among content delivery architectures. In: IEEE Telecommunications Internet and Media Techno Economics (CTTE), pp 1–6
18. Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York
19. Gkantsidis C, Karagiannis T, Vojnovic M (2006) Planet scale software updates. In: ACM SIGCOMM
20. Gunaratne C, Christensen K, Nordman B (2005) Managing energy consumption costs in desktop pcs and lan switches with proxying, split tcp connections, and scaling of link speed. Int J Netw Manag 15:297–310

21. Gupta M, Singh S (2003) Greening of the Internet. In: SIGCOMM
22. Hlavacs H, Weidlich R, Treutner T (2012) Energy efficient peer-to-peer file sharing. J Supercomput 62(3):1167–1188
23. Ilyas MS, Raza S, Chen CC, Uzmi ZA, Chuah CN (2014) RED-BL: Evaluating dynamic workload relocation for data center networks. Commun Netw 72:140–155
24. Intel: Enhanced intel speedstep technology for the intel pentium m processor. Intel White Paper 301170-001 (2004)
25. Krishnaswamy D, Krishnan R, Lopez D, Willis P, Qamar A (2015) An open NFV and cloud architectural framework for managing application virality behaviour. In: IEEE Consumer Communications and Networking Conference (CCNC), 2015, pp 746–754
26. Kumar R, Ross K (2006) Peer-assisted file distribution: The minimum distribution time. In: IEEE Workshop on Hot Topics in Web Systems and Technologies (HOTWEB 06), pp 1–11
27. Labovitz C, Iekel-Johnson S, McPherson D, Oberheide J, Jahanian F (2010) Internet inter-domain traffic. In: ACM SIGCOMM
28. Langner T, Schindelhauer C, Souza A (2011) Optimal file-distribution in heterogeneous and asymmetric storage networks. SOFSEM 2011: Theory and Practice of Computer Science pp 368–381
29. Lawey AQ, El-Gorashi T, Elmirghani J. M (2012) Energy-efficient peer selection mechanism for bittorrent content distribution. In: IEEE Global Communications Conference (GLOBECOM), pp 1562–1567
30. Lee U, Rimac I, Kilper D, Hilt V (2011) Toward energy-efficient content dissemination. IEEE Network Magazine 25(2):14–19
31. Lingjun M, Tsang P, Lui K (2009) Improving file distribution performance by grouping in peer-to-peer networks. IEEE Trans Netw Serv Manag 6(3):149–162
32. Malatras A, Peng F, Hirsbrunner B (2012) Energy-efficient peer-to-peer networking and overlays. Handbook on Green Information and Communication Systems
33. Mehyar M, Gu W, Low S, Effros M, Ho T (2007) Optimal strategies for efficient peer-to-peer file sharing. In: Acoustics, Speech and Signal Processing, ICASSP, vol. 4
34. Mundinger J, Weber R, Weiss G (2008) Optimal scheduling of peer-to-peer file dissemination. J Sched 11(2):105–120
35. Nedevschi S, Chandrashekar J, Liu J, Nordman B, Ratnasamy S, Taft N (2009) Skilled in the art of being idle: reducing energy waste in networked systems. In: NSDI
36. Nordman B, Christensen KJ (2009) Greener PCs for the enterprise. IT Professional 11(4):28–37
37. Ozkasap O, Cem E, Cebeci S. E, Koc T (2014) Flat and hierarchical epidemics in P2P systems: Energy cost models and analysis. Futur Gener Comput Syst 36:257–266
38. Pickavet M, Vereecken W, Demeyer S, Audenaert P, Vermeulen B, Develder C, Colle D, Dhoedt B, Demeester P (2008) Worldwide energy needs for ICT: The rise of power-aware networking. In: ANTS
39. Restrepo J, Gruber C, Machuca C (2009) Energy profile aware routing. In: Communications Workshops, IEEE ICC., pp 1–5
40. Sanghavi S, Hajek B, Massoulie L (2007) Gossiping with multiple messages. IEEE Trans Inf Theory 53(12):4640–4654
41. Sucevic A, Andrew L, Nguyen T (2011) Powering down for energy efficient peer-to-peer file distribution. ACM Sigmetrics Workshops, GreenMetrics
42. Trunfio P (2015) A two-layer model for improving the energy efficiency of file sharing peer-to-peer networks. Concurrency and Computation: Practice and Experience 27(13): 3166–3183
43. Tsang PS, Meng X, Lui KS (2010) A novel grouping strategy for reducing average distribution time in P2P file sharing. In: IEEE ICC
44. Valancius V, Laoutaris N, Massoulié L, Diot C, Rodriguez P (2009) Greening the Internet with nano data centers. In: ACM CoNEXT