Towards Efficient Operation of Internet Data Center Networks: Joint Data Placement and Flow Control for Cost Optimization

Abdallah Khreishah[1], Jacob Chakareski[2], Ammar Gharaibeh[1], Issa Khalil[3], Ali Diabat[4]

Abstract

The problem of cost-efficient operation of data center networks used to deliver **file sharing** services is studied. The aggregate costs are split into server-load-related and link-load-related shares. Thus, the problem of interest is formulated as one of joint data placement and flow control, and mixed integer-linear programming is used to compute the optimal solution. The high complexity of the latter motivated us to design two additional sets of strategies, based on data coding and heuristics, respectively. With coding, a distributed algorithm for the problem is developed. In the simulation experiments, carried out based on actual data center information, network topology and link cost, as well as electricity prices, the advantages of data coding, in particular in the context of multicast, and the impact of different factors such as the network topology and service popularity, on the total cost incurred by all considered strategies, are examined. Network coding with multicast is shown to provide cost savings in the order of 30-80%, depending on the specific context under consideration, relative to the other optimization strategies and heuristic methods examined in this work.

1. Introduction

A defining landmark of the present Internet are data center networks - global distributed systems featuring multiple geographically dispersed data centers and high speed data trunks between them. They are used to deliver a variety of services to increasingly bigger client populations online. The two most recognized characteristics of such systems are their sheer scale and power demand [1].

Email addresses: `abdallah@njit.edu` (Abdallah Khreishah), `jacob@ua.edu` (Jacob Chakareski), `amg54@njit.edu.` (Ammar Gharaibeh), `ikhalil@qf.org.qa` (Issa Khalil), `adiabat@masdar.ac.ae` (Ali Diabat)

[1]New Jersey Institute of Technology
[2]University of Alabama
[3]Qatar Computing Research Institute
[4]Masdar Institute of Science and Technology

Therefore, cost efficient operation of data center networks is a necessity. There are two types of cost that an Online Service Provider (OSP) operating a data center network needs to bear. The first one is the data transport cost associated with delivering the services to a multitude of demanding customers. The second one is the operating cost of the data centers that is associated with their actual location in the network. As the two cost items are interrelated, optimizing the network operation solely based on one, may lead to very inefficient performance relative to the case when the aggregate cost is taken into consideration. At the same time, the OSP needs to ensure that the services delivered over the network maintain their corresponding performance requirements at the end user. Traditionally, network traffic engineering to date has exclusively focused on load balancing across multiple internal paths in transit ISP networks [2]. Only in the domain of multihoming[5], we have witnessed interest in minimizing the transport cost of the customer [3] or the average latency of its data flows, for a given cost constraint [4]. Other studies try to minimize the network operational costs or the average latency by caching the files at different locations of the network [5, 6, 7, 8, 9]. The most closely related work is [10] that studies the joint data center placement and flow control in online service provider networks, with the goal of minimizing the overall operational cost of such networks, for the given performance guarantees associated with each service. Relative to [10], various unicast and multicast scenarios where the data is network-coded and the server load is taken into consideration are examined.

The work in [11] studies the problem of joint request mapping and response routing for cloud services running on geographically distributed data centers. The objective is to minimize the total costs (electricity and bandwidth costs) of serving clients' requests. Relative to [11], this work targets different types of cloud applications (i.e. file sharing), which do not have strict delay requirements. Moreover, various unicast and multicast scenarios are considered in this paper.

The work in [12] targets online services, such as search and instant messaging, that have strict delay requirements, and tries to find a curve, where each point in the curve represents a trade-off between the performance metric, which is Round Trip Time (RTT), and the operational costs of the Online Service Provider (OSP). Relative to [12], this paper considers file sharing services, which do not have strict delay requirements. Moreover, the work in [12] does not consider the operational costs of data centers.

The work in [13] proposes a two-stage optimization problem. The first stage is an admission control stage, where requests are selectively admitted to maximize the revenue collected by the Cloud Data Center (CDC) provider. The revenue of a request is modeled as a utility function that depends on the average response time. Given the admitted requests from the first stage, the second stage decides

[5] A customer is connected to the Internet via multiple ISPs.

which data center will satisfy the request and which path is chosen to deliver the response in order to minimize the total energy and bandwidth cost. Relative to [13], this work targets different types of cloud applications (file sharing) that do not have strict delay requirements. Moreover, the authors in [13] consider that the bandwidth cost of different links belonging to the same Internet Service Provider is the same, while this work considers that the cost of traversing a link depends on the bandwidth and the length of the link.

Other related work includes [14] that considers incorporating renewable energy sources as prospective suppliers of electricity for data center networks and the demand markets that therefore need to be stimulated, in order for the former to become a viable alternative in this context. [15, 16] study the advantages of network coding versus data replication in distributed content distribution systems.

To illustrate the benefits of our approach, assume that our data center is distributed over three sites that can be accessed over the Internet. Assume that they are located in Tokyo, Dublin, and Seattle (these cities could be changed to any other three)[6]. Fig. 1 represents our system model in which a distribution center decides the allocation of service requests to data centers. Assume that the three sites are powered via solar energy. However, if solar energy is not available and the data center is running, then it will use brown energy. In a typical summer day, it is fairly reasonable to assume that at any given time the solar energy will be available at two of the sites, and not available at the third one. Assume that a file of 2GB, is to be available at any given time through the three sites. Assume for simplicity that each site can store only 1GB. The objective here is to minimize the use of brown energy. Assume that the available space for this file at each site is 1GB. One approach to distribute this file over the data centers is to keep the first half of the file (A) on one site, say Tokyo, the second half of the file (B) on the second site, say Dublin, and to replicate either the first half of the file (A) or the second half (B) at Seattle. In this way, the total brown energy consumption would be $\frac{1}{3} \times BEE$, where BEE is the brown energy expenditure during a day per GB of data. That is because, assuming we chose A to be at Seattle, the Dublin site that contains B has to be powered all of the time, and in $\frac{1}{3}$ of the time it uses brown energy. With coding, instead of storing either A or B at Seattle, a coded packet A $\otimes$ B is stored, where $\otimes$ is the bitwise XOR operation. Therefore, at any given time accessing any two of the sites is sufficient to retrieve the whole file. Therefore, the brown energy use in this case will be zero. Fig. 2 represents our approach. The same example can be used to minimize the total energy use when a multi-electricity environment is present, in which each site is powered by a different provider, and the cost of electricity at these sites is different and changing over time. In this case, the site where the cost of electricity is the maximum among the different sites is

[6]This is a fairly realistic example, see for example Microsoft's Azure cloud that is distributed over three continents.

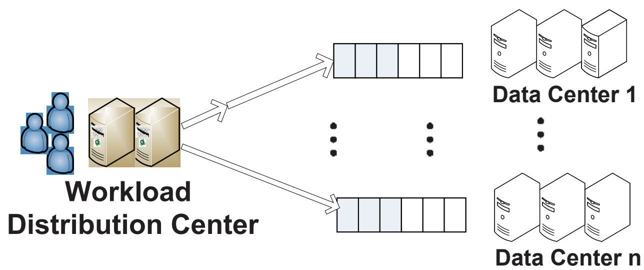
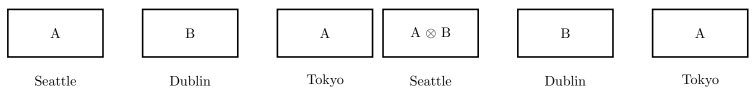


Figure 1: System Model.



(a) Non-coded solution (b) Coded solution

Figure 2: A representation of the example discussed above. Squares represent data centers. When using coding, accessing any two data centers can recover the whole file, which reduces brown energy consumption.

put to sleep [**17, 18, 19, 20**], and the other sites are turned on, if coding is used. In order to do that without coding, the whole file must be stored in the three sites which is expensive. The same approach can be used to minimize the energy used to cool the data centers, as it will change over time depending on the temperature, the pressure, and other factors.

The main contributions of our paper can be summarized as follows:

- Multiple formulations of the problem under consideration are derived that examine the impact of network coding, multicast, deterministic and stochastic settings, and decentralized operation.

- Network coding is demonstrated to reduce the overall system cost and the problem's complexity (NP-complete to polynomial).

- In the multi-cast case, the problem can be solved in polynomial time with very good performance, if network coding is performed in the intermediate nodes.

- A distributed algorithm for the problem with network coding is developed and its convergence to optimality is proved.

- The reduction of overall cost under network coding is shown through simulations, and the effect of different factors on the overall cost of all schemes are investigated.

The rest of the paper is organized as follows. First, some preliminaries are covered in Section 2. Then, the first formulation of the problem under consideration is presented in Section 3. Next, multiple extensions of the problem under consideration are designed that include network coding, multicast, and step power consumption in Section 4. A distributed algorithm for the network coding formulation based on the proximal method and the Lagrange multiplier method is presented in Section 5. Heuristic algorithms that compute the optimal solution at lower complexity are designed in Section 6. Experimental evaluation of the various optimization scenarios and algorithms under consideration is carried out in Section 7. Finally, the paper is concluded in Section 8.

2. Preliminaries

The network is modeled as a directed graph $G = (V, E)$, where V is the set of nodes, and E is the set of edges that represent the links between the nodes. The nodes can represent the data centers, the routers, or the host nodes. Let S_k represent the size of the k-th file to be stored in distributed storage across the data centers, and let H_k represent the sending rate of the k-th file, which is assumed to be equal to the receiving rate. Each client can request a file or not. For that I_{ik} is used, such that if $I_{ik} = 1$, this means that the k-th file is being requested by the i-th node. Otherwise $I_{ik} = 0$. Let P_{ik}^{j} be a binary variable that is equal to one, if node i stores block j for file k, and is zero, otherwise. Let R_i^k represent the number of data blocks affiliated with file k stored at node i that represents one of the data centers, D_i the total amount of storage allocated at node i, and symbol d_i the total traffic load at node i. The total amount of flow at link l is represented by f_l, and X_l^{ik} is used to represent the rate of flow at link l for the k-th file destined for node i. Let $I_i^k(u)$ represent the set of previous hop nodes of node u on the shortest path from the location of the k-th file to node i, and let $O_i^k(u)$ represent the set of next-hop nodes of node u on the shortest path from the location of the k-th file to node i. Table 1 compiles the major symbols used in the paper.

3. Basic Formulation

First, a basic formulation of the problem is considered, where the file demand across the nodes of the network is deterministic, and no network coding is applied on the data affiliated with any file. In this case, C_1 represents the cost associated with the traffic load that the data center servers need to meet. C_2 represents the cost associated with the amount of storage allocated at each server. Finally, C_3 is the cost of transporting data between data centers and consumers. As explained earlier, we are interested in minimizing the overall cost of operating the network, which formally can be described as

$$\min_{X_l^{ik}, P_{ik}^{j}} \sum_i [C_1(d_i) + C_2(D_i)] + \sum_l C_3(f_l) \tag{1}$$

$$\text{s.t.} \sum_{l \in I_i^k(u)} y_{lj}^{ik} - \sum_{l \in O_i^k(u)} y_{lj}^{ik} = H_k I_{ij} 1_{u=i} - H_k P_{uk}^j, \quad \forall u, i, j, k \tag{2}$$

$$\sum_j y_{lj}^{ik} = X_l^{ik} \quad \forall l, i, k \tag{3}$$

$$\sum_{j=1}^{S_k} P_{ik}^j = R_i^k \quad \forall i, k \tag{4}$$

$$\sum_k R_i^k = D_i \quad \forall i \tag{5}$$

$$\sum_i \sum_k X_l^{ik} = f_l \quad \forall l \tag{6}$$

$$\sum_u \sum_k \Big[\sum_{l \in I_u^k(i)} X_l^{uk} + \sum_{l \in O_u^k(i)} X_l^{uk} \Big] = d_i \quad \forall i, \tag{7}$$

where X_l^{ik} and P_{ik}^j represent our variables of control. Constraint (2) represents the flow conservation at each node at the block level. Constraint (3) states that the flow rate of a file destined for a certain node on a certain link is equal to the sum of the rates of the individual blocks of that file that are destined for the same node and use the same link. Constraint (4) states that the amount of storage for a file at a certain data center equals the number of stored blocks, while constraint (5) states that the storage at each data center is equal to the aggregated storage of all files to be stored at that data center. Constraint (6) states that the flow at each link is equal to the aggregated flow of all files that use that link. Constraint (7) states that the traffic at each data center is equal to the aggregated flows that enter or exit that data center. To solve this optimization problem, the popular optimization software package CPLEX [21] is used.

Note that the basic formulation presented above is a mixed integer linear program (MILP) which is generally very complex to solve. This high complexity motivated us to design two additional sets of strategies. The first set is based on Network Coding, which reduces the basic formulation to a Linear Program (LP) that can be solved in polynomial time, and achieves a lower cost than that achieved through solving the MILP formulation. The second set of strategies is based on heuristics, which are very simple to solve but at the expense of higher cost. The decision making can be realized using a centralized controller, such as the centralized controller used in Software Defined Networking [22, 23, 24]. Moreover, a distributed way of solving the network coding case is presented such that the decision making is distributed among the data centers and the Internet Service Providers.

4. Extensions

In the following, multiple extensions of the basic problem formulated in Section 3 are considered that may correspond to various variations of the scenario considered by (1) to (7) that may arise in practice. First, an extension where the individual blocks of the file are coded is considered; however, they are delivered over unicast connections to the clients. Then, a scenario where intermediate nodes (routers) mix the files associated with different flows using network coding is considered, to improve throughput. Hence, all files are delivered in a multicast fashion. Next, a scenario where the data is not coded at all, and each file is either fully stored at a data center or not at all is examined. Furthermore, a scenario is studied where operating a data center comprises a load-independent cost factor that arises whenever the data center needs to be powered on [25]. Finally, a dynamic scenario where the operation of the data center network is considered over a period of time is studied. Specifically, a fixed data assignment setup is studied, where the data allocation at different network nodes will not change over time, as opposed to the demand. To conserve space, in the following subsections, only the differences of each scenario relative to the basic problem, in terms of formulation are highlighted.

4.1. Formulation with Network Coding (unicast)

The constraints (2) to (4) are replaced with the following constraint:

$$\sum_{l \in I_i^k(u)} X_l^{ik} - \sum_{l \in O_i^k(u)} X_l^{ik} = H_k I_{ik} 1_{u=i} S_k - H_k R_u^k \quad \forall u, i, k \tag{8}$$

Constraint (8) represents the flow conservation at each node, where receiving any S_k linearly independent blocks is sufficient to retrieve the whole file. It has been shown in [26] that with a moderate finite field size, typically 16, the different coded blocks will be linearly independent with a very high probability.

4.2. Formulation with Network Coding (Multicast)

Relative to unicast, constraints (6) to (7) are replaced with:

$$X_l^{ik} \leq Z_l^k \qquad \forall l, \forall k, \forall i \tag{9}$$

$$\sum_k Z_l^k \leq f_l \qquad \forall l \tag{10}$$

$$\sum_k \Big[\sum_{l \in I^k(i)} Z_l^k + \sum_{l \in O^k(i)} Z_l^k \Big] = d_i \quad \forall i \tag{11}$$

Constraints (9) to (11) support multicasting through data coding at intermediate nodes, as in [27]. Here, Z is a slack variable.

4.3. No Coding, file stored entirely or not at all

Here, a formulation without coding is presented, where the file is either entirely stored at a data center or not at all. The notation is kept the same as in the deterministic case, except the definition of R_i^k, where $R_i^k = 1$, if node i has all the blocks of file k, and $R_i^k = 0$, otherwise. Then, constraints (2) to (5) are replaced with:

$$\sum_{l \in I_i^k(u)} X_l^{ik} - \sum_{l \in O_i^k(u)} X_l^{ik} = H_k I_{ik} 1_{u=i} S_k - H_k R_u^k S_k, \ \forall u, i, k \tag{12}$$

$$\sum_k R_i^k S_k = D_i \quad \forall i \tag{13}$$

Constraint (12) represents the flow conservation law for node i. Constraint (13) states that the file is either stored entirely or not at all.

4.4. Step Power Consumption

Let $a_i = 1$, if data center i is turned on, and $a_i = 0$, otherwise. Let b_i represent the baseline power consumption of an operating data center. Then, the objective function is changed to:

$$\min_{X_l^{ik}, P_{ik}^j} \Big[\sum_i [C_1(d_i) + C_2(D_i) + a_i b_i] + \sum_l C_3(f_l)\Big]$$

and a new constraint is added:

$$\text{s.t. } f_l \leq a_i C_l \quad \forall i, l \in I(i), O(i) \tag{14}$$

where C_l represents the link l capacity. Constraint (14) states that if a data center is off, then we do not use the links associated with that data center.

4.5. Static Data Assignment

Here, we are interested in minimizing the operating cost over a horizon of time, for static data placement. That is,

$$\min_{X_l^{ik}, P_{ik}^j} \sum_t \Big[\sum_i [C_1(t, d_i(t)) + C_2(t, D_i)] + \sum_l C_3(t, f_l(t))\Big]$$

s.t.

$$\sum_{l \in I_i^k(u)} y_{lj}^{ik}(t) - \sum_{l \in O_i^k(u)} y_{lj}^{ik}(t) = H_k I_{ik}(t) 1_{u=i} - H_k P_{uk}^j \tag{15}$$

$\forall u, i, k, j, t$

$$\sum_j y_{lj}^{ik}(t) = X_l^{ik}(t) \qquad \forall l, \forall i, \forall k, \forall t \tag{16}$$

$$\sum_{j=1}^{S_k} P_{ik}^j = R_i^k \qquad \forall i, \forall k \tag{17}$$

$$\sum_k R_i^k = D_i \qquad \forall i \tag{18}$$

$$\sum_i \sum_k X_l^{ik}(t) = f_l(t) \qquad \forall l, \forall t \tag{19}$$

$$\sum_u \sum_k \Big[\sum_{l \in I_u^k(i)} X_l^{uk}(t) + \sum_{l \in O_u^k(i)} X_l^{uk}(t) \Big] = d_i(t) \qquad \forall i, \forall t \tag{20}$$

In this formulation, the variables related to storage at a data center (P_{ik}^j, R_i^k, D_i) do not change with time, since the data allocation needs to be fixed. All other variables are related to the traffic flow, which can change over time depending on the time-changing demand.

4.6. Stochastic Case Formulation without Network Coding

In the stochastic case, there is a set $\mathcal{S} = \{\hat{S}_0, \hat{S}_1, \ldots\}$ of states that the demand can exhibit as it evolves dynamically. Let $\mathcal{P}$ denote the stochastic matrix that describes the transition between demand states. Let $\mathcal{T} = \{t_0, t_1, \ldots, t_{T-1}\}$ denotes the set of time instances at which we are interested in solving the problem. We are interested in minimizing the average cost of all states, where the cost of a state $\hat{S}$ is given by $C(\hat{S}) = \sum_i \big[C_1(d_i) + C_2(D_i)\big] + \sum_l C_3(f_l)$.

$$\min \sum_{m=0}^{T-1} \sum_{\hat{S} \in \mathcal{S}} \mathcal{P}^{(m)}(\hat{S} | \hat{S}_0) C(\hat{S})$$

subject to

$$\sum_{l\in I_i^k(u)} y_{lj}^{ik} - \sum_{l\in O_i^k(u)} y_{lj}^{ik} =$$

$$\sum_{\hat{S}\in\mathcal{S}} \mathcal{P}^m(\hat{S}|\hat{S}_0)H_kQ_{ij}(t_m)1_{u=i} - H_kP_{uj}^k \quad \forall u,i,j,k,t_m \tag{21}$$

$$\sum_j y_{lj}^{ik} = X_l^{ik} \quad \forall l,i,k \tag{22}$$

$$\sum_{j=1}^{S_k} P_{ik}^j = R_i^k \quad \forall i,k \tag{23}$$

$$\sum_k R_i^k = D_i \quad \forall i \tag{24}$$

$$\sum_i \sum_k X_l^{ik} = f_l \quad \forall l \tag{25}$$

$$\sum_u \sum_k \Big[\sum_{l\in I_u^k(i)} X_l^{uk} + \sum_{l\in O_u^k(i)} X_l^{uk} \Big] = d_i \quad \forall i \tag{26}$$

$$\sum_{\hat{S}\in\mathcal{S}} [\mathcal{P}^{(m)}(\hat{S}|\hat{S}_0) \sum_i Q_{ij}(t_m)] \geq \kappa \sum_{\hat{S}\in\mathcal{S}} [\mathcal{P}^{(m)}(\hat{S}|\hat{S}_0) \sum_i p_{ij}^{\hat{S}}] \quad \forall j, t_m \tag{27}$$

where constraint (21) represents the flow conservation constraint, and constraint (27) ensures that at least aκ fraction of the users are getting their requested services.

4.7. Stochastic Case Formulation with Network Coding

Replace constraints (21)-(23) with the following

$$\sum_{l\in I_i^k(u)} X_l^{ik} - \sum_{l\in O_i^k(u)} X_l^{ik} =$$

$$\sum_{\hat{S}\in\mathcal{S}} \mathcal{P}^m(\hat{S}|\hat{S}_0)H_kS_kQ_{ij}(t_m)1_{u=i} - H_kS_kR_u^k \quad \forall u,i,k,t_m \tag{28}$$

5. Distributed Algorithm for Network Coding Case

In this section, an adaptive and distributed algorithm for the network coding case of our problem is developed. This is motivated by the following reasons:

- The linear program of the network coding case is a convex program
- Using the dual approach [28, 29], the dual program is separable [30].

The challenge in this method is that the objective function is a linear function and hence is not strictly convex. This may cause the algorithm to oscillate. To alleviate this challenge, the proximal method [30] is used to convert the objective function to a strictly convex function. This is done by introducing the auxiliary variables $\hat{d}_i$, $\hat{D}_i$, $\hat{f}_l$, and changing the objective function to the following:

$$\min \sum_i (C_1(d_i)+C_2(D_i))+\sum_l C_3(f_l)+\sum_i \frac{\hat{\gamma}_i}{2}(d_i-\hat{d}_i)^2+\sum_i \frac{\gamma_i}{2}(D_i-\hat{D}_i)^2+\sum_l \frac{\gamma_l}{2}(f_l-\hat{f}_l)^2$$

where γ_i, $\hat{\gamma}_i$, and γ_l are positive constants.

The Lagrange cost for the network coding case formulation can be written as follows:

$$\begin{aligned}
L(\vec{X},\vec{R},\vec{D},\vec{f},\vec{d},\vec{\lambda},\vec{\hat{d}},\vec{\hat{D}},\vec{\hat{f}}) &= \sum_i (C_1(d_i)+C_2(D_i)) + \sum_l C_3(f_l) \\
&+ \sum_i \frac{\hat{\gamma}_i}{2}(d_i-\hat{d}_i)^2 + \sum_i \frac{\gamma_i}{2}(D_i-\hat{D}_i)^2 + \sum_l \frac{\gamma_l}{2}(f_l-\hat{f}_l)^2 \\
&+ \sum_{u,i,k} \lambda_{ui}^k [\sum_{l\in O_i^k(u)} X_l^{ik} - \sum_{l\in I_i^k(u)} X_l^{ik} \\
&+ H_k I_{ik} 1_{u=i} S_k - H_k R_u^k] \\
&+ \sum_i \lambda_i [\sum_k R_i^k - D_i] \\
&+ \sum_l \lambda_l [\sum_i \sum_k X_l^{ik} - f_l] \\
&+ \sum_i \hat{\lambda}_i \Big[\sum_u \sum_k [\sum_{l\in I_u^k(i)} X_l^{uk} + \sum_{l\in O_u^k(i)} X_l^{uk}] - d_i\Big]
\end{aligned}$$

The distributed algorithm is presented in Algorithm 1. Note that $\beta_i, \hat{\beta}_i, \beta_{ui}^k, \beta_l$ are the step sizes of the gradient projection method. **The proof of the algorithm's convergence to the optimal solution is presented in Theorem 1.**

Theorem 1. *Algorithm 1 converges to the optimal solution of the formulation presented in Section 4.1, if the step sizes satisfy the following conditions(the subscripts of the step sizes are dropped as these conditions have to be met for all step sizes):*

- $\beta > 0$.
- $\lim_{t\to 0} \beta(t) = 0$.
- $\sum_{t=1}^{\infty} \beta(t) = \infty$.

Proof. The proof is similar to what is presented in [31] □

Algorithm 1 Distributed Algorithm

Initialize all Lagrange multipliers to 0.

At the t-th iteration, perform the following steps sequentially for $m = 0, 1, \ldots, M$

Data center i updates the values of $\lambda_i(t, m+1), \hat{\lambda}_i(t, m+1)$ as follows:

$\lambda_i(t, m+1) = \left[\lambda_i(t, m) + \beta_i[\sum_k R_i^k(t, m) - D_i(t, m)]\right]^+$

$\hat{\lambda}_i(t, m+1) = \left[\hat{\lambda}_i(t, m) + \hat{\beta}_i[\sum_u \sum_k \left[\sum_{l \in I_u^k(i)} X_l^{uk}(t, m) + \sum_{l \in O_u^k(i)} X_l^{uk}(t, m)\right] - d_i(t, m)]\right]^+$

Link l updates the value of $\lambda_l(t, m+1)$ as follows:

$\lambda_l(t, m+1) = \left[\lambda_l(t, m) + \beta_l[\sum_i \sum_k X_l^{ik}(t, m) - f_l(t, m)]\right]^+$

Relay node u updates the value of $\lambda_{ui}^k(t, m+1)$ as follows:

$\lambda_{ui}^k(t, m+1) = \left[\lambda_{ui}^k(t, m) + \beta_{ui}^k[\sum_{l \in O_i^k(u)} X_l^{ik}(t, m) - \sum_{l \in I_i^k(u)} X_l^{ik}(t, m) + H_k I_{ik} 1_{u=i} S_k - H_k R_u^k(t, m)]\right]^+$

Let $\vec{X}(t, m+1), \vec{R}(t, m+1), \vec{D}(t, m+1), \vec{f}(t, m+1), \vec{d}(t, m+1) = \arg\min L(\vec{X}, \vec{R}, \vec{D}, \vec{f}, \vec{d}, \vec{\lambda}(t, m+1), \hat{\vec{d}}(t), \hat{\vec{D}}(t), \hat{\vec{f}}(t))$

Let $\vec{\lambda}(t+1, 0) = \vec{\lambda}(t, M)$

Set:

$\hat{\vec{d}}(t+1) = \vec{d}(t, M)$

$\hat{\vec{D}}(t+1) = \vec{D}(t, M)$

$\hat{\vec{f}}(t+1) = \vec{f}(t, M)$

$\vec{d}(t+1) = \vec{d}(t, M)$

$\vec{D}(t+1) = \vec{D}(t, M)$

$\vec{f}(t+1) = \vec{f}(t, M)$

$\vec{R}(t+1) = \vec{R}(t, M)$

$\vec{X}(t+1) = \vec{X}(t, M)$

Note that the variables in the algorithm can be updated using local information.

6. Heuristic Algorithms

The basic formulation mentioned in Section 3 is a mixed integer linear program, which is NP hard. Network coding reduces the problem's complexity to that of a linear program which can be solved in polynomial time. In the following, a number of heuristic techniques for computing a data placement and flow control solution at lower complexity are introduced.

First, some necessary notation is provided below. In particular:

- A: An adjacency matrix, where a value in the i-th row and j-th column represents the cost of traversing the direct link between nodes i and j.
- W: Shortest path cost matrix: It comprises the cost of the shortest path between nodes i and j in the network, where the shortest paths are computed according to Dijkstra's algorithm.
- $S = (S_1, \ldots, S_M)$: Number of blocks in each file, we have M files.
- N: Number of data centers.
- C: Number of clients.
- $I_{ik} = \begin{cases} 1 & \text{if client } i \text{ is requesting file } k \\ 0 & \text{otherwise} \end{cases}$

6.1. No Coding, One Duplication

Input: W, I
for $k = 1$ to M
    Initialize B_u to an all zero $N \times 1$ array
    for $u = 1$ to N
        $B_u = 0$
        for $i = 1$ to C
            if $I_{ik} = 1$
                $B_u = B_u + W(u, i)$
            end if
        end for
    end for
    Find $u = argmin(B_u)$
    Store file k on server u
end for

Each file is stored at the server that is closest to the demanding nodes (the server with the minimum B_u). Specifically, for each file, the aggregated link cost on all shortest paths from each server to all clients that requested the file is computed. The file is then stored at the server with the minimum cost. The total cost then is computed as the sum of the storage and traffic costs on the chosen servers and the cost of the links on the shortest path from the servers to the demanding clients, over all files. The complexity of this algorithm arises from the loops and the calculation of the shortest paths, which is $\mathcal{O}(M*N*C+(N+C)^3)$.

6.2. No Coding, 2 Duplications

Input: W, I
for $k = 1$ to M
  Initialize matrix B_u^j to an all zero $N \times N$ matrix
  for $u = 1$ to $N-1$
    for $j = u+1$ to N
      for $i = 1$ to C
        if $I_{ik} = 1$
          $B_u^j = B_u^j + \min(W(u,i), W(j,i))$
        end if
      end for
    end for
  end for
  Find u, j that gives the minimum B_u^j
  Store file k on servers u, j
end for

Each file is stored at the two servers that are closest to the demanding nodes. Specifically, for each file, two servers are taken at a time and the aggregated link cost on the shortest path from each client to the closest of the two servers is computed. The file is stored on both servers that have the minimum aggregated cost, and each client will get the file from the closest of the two servers. The total cost is computed as in section A. The complexity of this algorithm is $\mathcal{O}(M*N^2*C+(N+C)^3)$

6.3. No Coding, File Division

Input: W, I, m
Divide each file into m divisions, $m \leq N$
for $k = 1$ to M
  Initialize B_u to an all zero $N \times 1$ array
  for $u = 1$ to N
    $B_u = 0$
    for $i = 1$ to C
      if $I_{ik} = 1$
        $B_u = B_u + W(u,i)$
      end if
    end for
  end for

Sort B_u in Ascending Order
Get servers u in the first m elements of B_u
Store each division of file k on m servers
end for

Each file will be divided into m divisions, and each division will be stored on one of the m servers closest to the demanding nodes. Here, the aggregated link cost on the shortest path from each server to the demanding clients is computed, as in Section A, but instead of storing the whole file on the closest server, the file is divided into m divisions and each part is stored at one of the m closest servers. The complexity of this algorithm is $\mathcal{O}(M * N * C + (N + C)^3)$

7. Evaluation

In Section 7.1, the performance of the following schemes are studied, via simulation experiments.

- **Extensions presented in Sections 4.1, 4.2, 4.3.**
- **Heuristics algorithms presented in Section 6.**
- **ADMM as adopted from [11].**
- **SweetSpot as adopted from [12].**

In particular, the effect of file popularity, the ratio of storage cost to transmission cost, the variance in storage cost, and the network topology, on the resulting energy cost of each scheme are investigated. In Section 7.2 the same simulations are repeated when the power step as presented in Section 4.4 is taken into account.

As proved in Theorem 1 in Section 5, the distributed algorithm presented in Algorithm 1 in Section 5 converges to the optimal solution of the network coding case presented in Section 4.1. Therefore, the simulation results for the network coding is representative of both the network coding extension presented in Section 4.1 and the distributed algorithm.

Note that in order to perform an evaluation of real systems, different clusters at geographically distributed locations need to be setup, and configuring a single cluster will not solve the problem. Therefore,trace-driven simulations are performed. All of the works targeting geographically distributed data centers perform trace-driven simulations such as [11, 12, 13].

7.1. Results for the Basic Formulation

7.1.1. Setup

The number of files considered in the simulations was 16, where each comprises a number of blocks ranging between 10 and 30, with a size of 1 GB per block. The electricity prices for 3 Google data centers located in TX, GA, and CA were adopted from [32]. The electricity prices for the rest of the data centers were taken as a weighted average depending on the distance from all three Google data centers. Power consumption in a data center was calculated as follows: A simple web search, which is around 35kB, consumes 0.0003 kWh including the power consumption from the cooling units [33]. Along with the block size and the electricity price, the power consumption cost is easily found. As for the link costs, $1 buys 2.7 GB bandwidth over a link of 100 kilometers as adopted from [34]. The traffic cost at each data center was taken as $0.12 per GB per month, as adopted from [35].

7.1.2. Topology Effect

The simulations are run for two different topologies. The first is the US backbone topology shown in Fig. 3, as reproduced from [36]. In this topology, the numbers in the circles represent the node IDs, while the numbers on the links represent their length in kilometers. The nodes with ID in {1,5,9,11,14,17,19,23} were picked as data centers. The rest of the nodes were considered as clients. The second topology is a random topology, where the nodes have coordinates drawn from a uniform distribution, and a link is established if the distance between two nodes is less than a certain threshold. The nodes representing the data centers are the same as those for the US backbone topology.

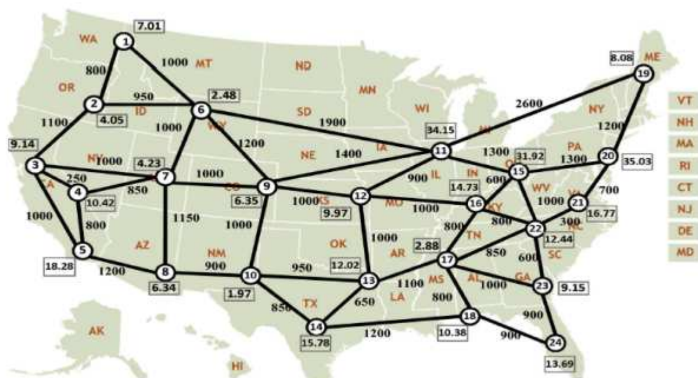


Figure 3: US Backbone Topology.

Table 2 summarizes the results of the simulation after normalizing them with the corresponding results for network coding under multicast. In particular, the table shows the aggregate cost increase in % incurred by a scheme, relative to network coding under multicast. From the results, it is observed that network topology and the location of data centers in the network affect the total energy cost incurred by all schemes.

7.1.3. Popularity Effect

Here, the impact of the popularity of each file among the requesting clients on the total cost is studied. The US backbone topology was chosen for this simulation. The topology was divided into 4 regions, East, Central, Mountain, and Pacific. Two different popularity schemes named Unary and Power Set were simulated. In Unary popularity, each file is considered to be popular in one region only. In Power Set popularity, some files were considered popular in one region, some in two regions, some in three regions, and the rest in all four regions. A file that is popular in a region has an 80% probability of being demanded, while unpopular files in a region have a 30% demand probability.

Table 3 summarizes our simulation findings. As in Table 2, the results represent the additional aggregate cost incurred by all schemes, with respect to network coding under multicast, in order to assess the benefits of the latter. It is seen that Coding performs better than the other schemes under consideration. It is also noted that the cost under Power Set popularity is higher than that under Unary popularity. That is because under Unary popularity, the demand for a file is concentrated in one region, and the related transmission cost will be lower, than when satisfying demands that are distributed over the whole network.

7.1.4. Ratio of Storage Cost to Transmission Cost Effect

To study the influence of the ratio of storage cost to transmission cost on the performance of all schemes, this quantity was varied between 10, 50, and 100, in the case of the US backbone topology under the Power Set popularity. The total cost of each scheme after normalizing with respect to network coding under multicast when the ratio is equal to 10 is shown in Fig. 4. When the ratio is small and the transmission cost is comparable to the storage cost, All File tends to duplicate a file near the demand locations instead of retrieving a part of the file from a distant location as in Coding. When the ratio increases and the storage cost becomes the dominant factor, Coding outperforms All File. **It is also shown in the figure that network coding under multicast can achieve cost savings up to 90% and 93% when compared to ADMM and SweetSpot, respectively. This is because ADMM and SweetSpot schemes do not optimize for data placement, and the files are stored at every data center. Thus as the storage cost increases, the cost of ADMM and SweetSpot schemes also increases.**

7.1.5. Storage Cost Variance Effect

In this simulation, instead of using the electricity prices, as explained in the beginning of this section, three random sets of prices were generated that have the same average value but differ in their variance. The variance was changed between 20%, 50%, and 80% of the chosen average value. The topology chosen for these experiments is the US backbone topology under the Power Set

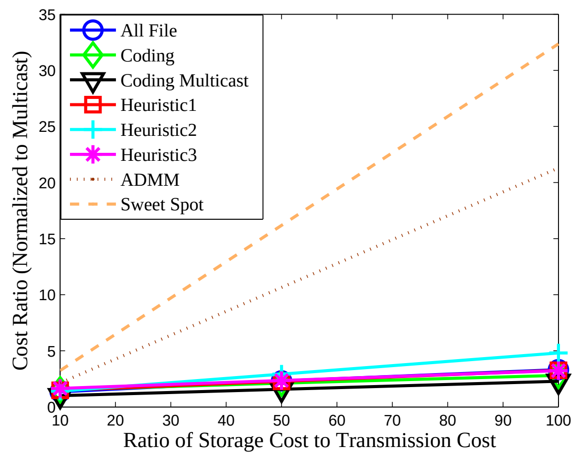


Figure 4: Storage Cost to Transmission Cost Ratio Effect.

popularity. The results, after normalizing with respect to the network coding under multicast when the variance is 20%, are shown in Fig. 5. When the variance of the storage cost is low, the storage cost values for all servers are close to each other, and the heterogeneous nature of the file demand becomes the deciding factor. When the variance is high, the heterogeneity of the storage cost becomes the dominant factor. In the middle, the storage cost heterogeneity and the file demand heterogeneity compensate for each other, and a lower ratio for the additional cost is observed. **The figure also shows that the total cost of ADMM and SweetSpot schemes can be up to 5 and 6 fold respectively when compared to network coding under multicast. This is because ADMM and SweetSpot schemes do not optimize for data placement.**

7.1.6. Number of Files Effect

In this simulation, the number of files that can be requested by the users is changed. The topology chosen for these experiments is the US backbone topology under the Power Set popularity. The results, after normalizing with respect to the network coding under multicast are shown in Fig. 6. It is seen from the figure that as the number of files increases, the total cost increases. The figure also shows that as the number of files increases, the benefits of using network coding under multicast becomes more apparent, as the cost savings over the All File scheme increase from 40%, when the number of files is low, to 52%, when the number of files is high. **Moreover, network coding under multicast can achieve cost savings of 75% and 83% over the ADMM and SweetSpot schemes respectively when the number of files is low, and 77% and 86% respectively when the number of files is high.**

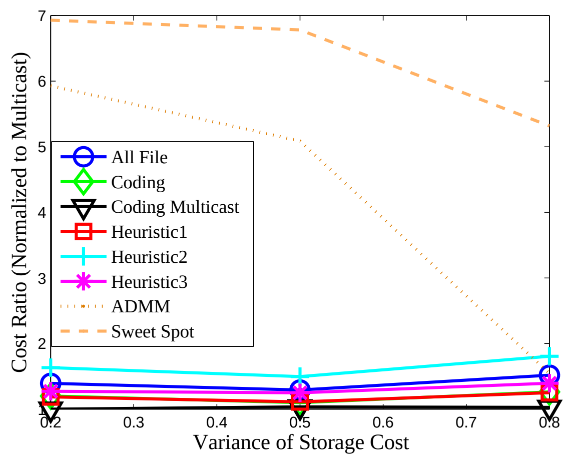


Figure 5: Storage Cost Variance Effect.

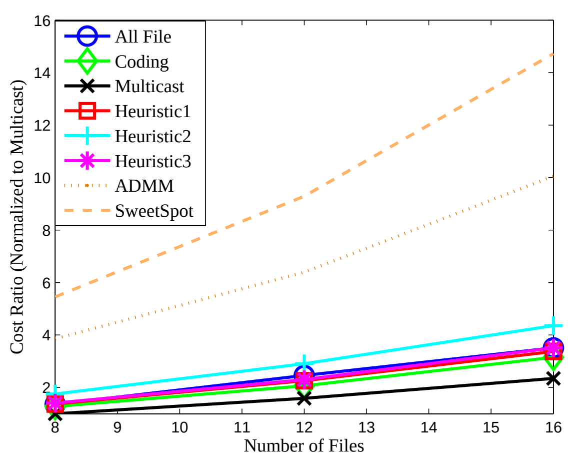


Figure 6: Effect of Number of Files.

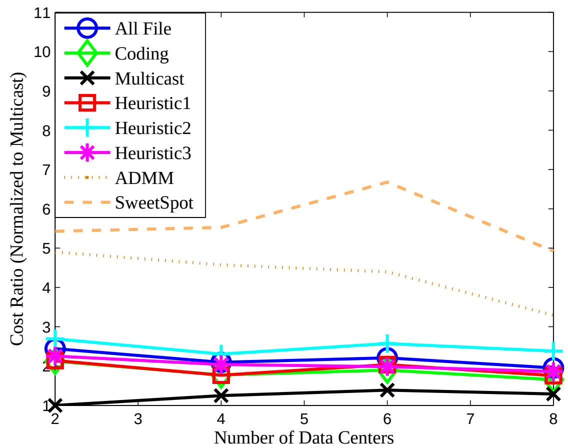


Figure 7: Effect of Number of Data Centers.

7.1.7. Number of Data Centers Effect

In this simulation, the effect of changing the number of data centers in the network is investigated. The topology used is the US backbone topology under the power set popularity. Fig. 7 shows the results for the simulation after normalizing with respect to the network coding under multicast. It is shown that there are points where a tradeoff between the storage cost and the transmission cost can be observed. **It is also observed that network coding can reduce the cost between 25% to 90% when compared to the other schemes.**

7.2. Results for the Power Step Formulation

7.2.1. Setup

The number of files considered in our simulations was 16, where each file comprises a number of blocks ranging between 10 and 30, with a size of 1 GB per block. Power consumption in a data center was calculated as follows: A simple web search, which is around 35kB, consumes 0.0003 kWh including the power consumption from the cooling units [33]. Along with the block size, the power consumption is easily found. The power consumption for transmission is assumed to be 0.2 kWh per 1 GB as adopted from [37]. The power consumed to activate a server is assumed to be 70 Watts [38]. Thus, a data center containing 100 servers requires 7 kW to be activated. All the results presented in this section are run using the US backbone topology under the power set popularity.

7.2.2. Ratio of Storage Power Consumption to Transmission Power Consumption Effect

In this simulation, the ratio between the power consumption of data storage and data transmission is varied. It is seen from Fig. 8 that when the power

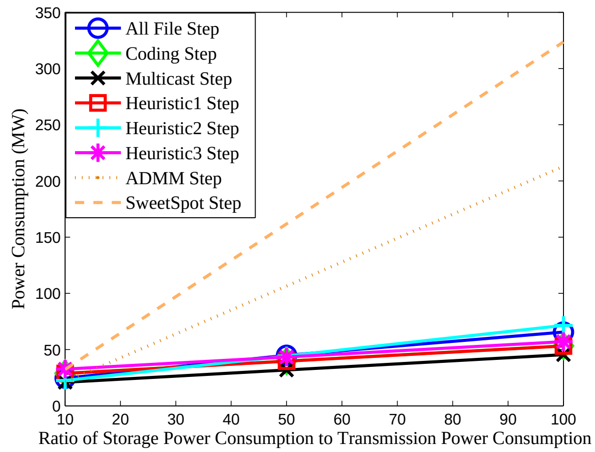


Figure 8: Storage Cost to Transmission Cost Ratio Effect for the Step Power Formulation.

consumption of data storage is lower than the power consumption of data transmission, All Files scheme consumes less power than the network coding under unicast scheme. As the ratio increases, the power consumption of data storage becomes the dominant factor and the network coding schemes consume less power than the All Files scheme. **It is also noted that the total power consumption of ADMM and SweetSpot schemes can be up to 4 and 7 fold respectively when compared to network coding under multicast scheme. This is because ADMM and SweetSpot schemes do not optimize for data placement. Therefore, as the power consumption required for storage increases, the power consumption of ADMM and SweetSpot increases.**

7.2.3. Storage Power Consumption Variance Effect

Here, the variance of the power consumption of data storage across the data centers is varied. It is observed from Fig. 9 that when the variance is high, all schemes tend to store the data in the data centers with low power consumption, thus achieving a lower power consumption relative to the case of low power consumption variance. Moreover, it is noted that the network coding under multicast scheme achieves between 19% to 65% reduction in power consumption over the All Files scheme. **It is also shown in the figure that network coding under multicast can achieve up to 80% and 81% reduction in power consumption over the ADMM and SweetSpot schemes, as ADMM and SweetSpot schemes do not optimize for data placement.**

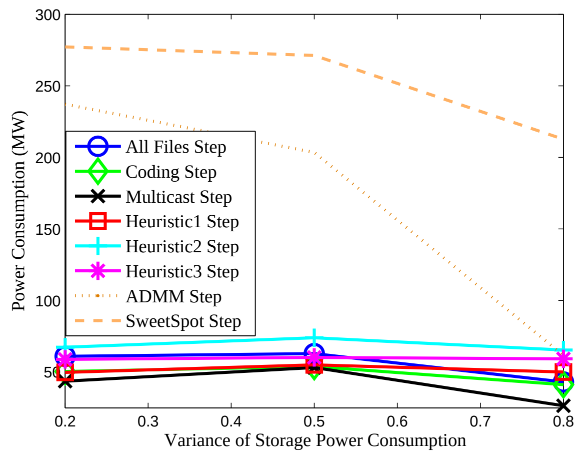


Figure 9: Storage Cost Variance Effect for the Step Power Formulation.

7.2.4. Number of Files Effect

Fig. 10 examines the impact of the number of files requested by the users. It is shown that as the number of files increases, the total power consumption increases. Moreover, it is observed that the power consumed by the network coding under multicast scheme is 44% less than the power consumption of the All File scheme, when the number of files is high. **It is also observed from the figure that network coding under multicast can achieve a reduction in power consumption of 33% and 60% over the ADMM and SweetSpot schemes respectively when the number of files is low, and 50% and 65% respectively when the number of files is high.**

7.2.5. Number of Data Centers Effect

Fig. 11 shows the results when the number of data centers is changed. From the figure, it is noted that as the number of data centers increases, the total power consumption decreases. This is because activating a data center closer to the users consumes less power than transmitting lots of data over greater distances. Moreover, it is observed that after a certain point, the benefit of activating additional data centers diminishes, since there are already enough data centers close to the users, and activating any additional data centers consumes more power than transmitting the data over short distances. Finally, it is noted that the network coding under multicast scheme can achieve between 30% to 90% reduction in power consumption over the All Files scheme**, between 55% to 66% reduction in power consumption over ADMM scheme, and between 58% to 75% reduction in power consumption over SweetSpot scheme**.

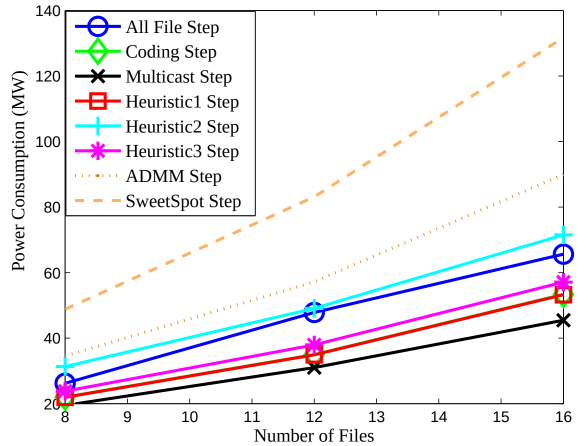


Figure 10: Effect of Number of Files for the Step Power Formulation.

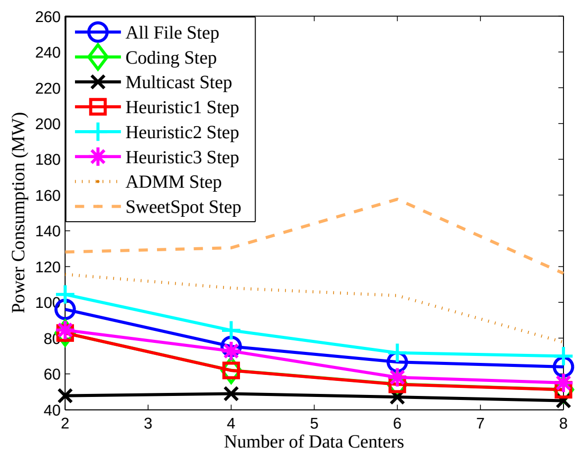


Figure 11: Effect of Number of Data Centers for the Step Power Formulation.

8. Conclusion

From the simulations, it is shown that network coding, both under unicast and multicast traffic, reduces the overall cost incurred by file storage and network traffic. The effect of different factors on the overall cost of all schemes is investigated. It is found that the location of data centers in the topology, the file popularity, and the ratio of storage cost to transmission cost have the biggest impact on the overall cost.

References

[1] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, B. Maggs, Cutting the electric bill for Internet-scale systems, in: Proc. SIGCOMM, ACM, 2009.

[2] M. Roughan, M. Thorup, Y. Zhang, Traffic engineering with estimated traffic matrices, in: Proc. Internet Measurement Conference, ACM SIGCOMM, Miami Beach, FL, USA, 2003, pp. 248–258.

[3] A. Akella, J. Pang, B. Maggs, S. Seshan, A. Shaikh, A comparison of overlay routing and multihoming route control, in: Proc. SIGCOMM, ACM, Portland, OR, USA, 2004, pp. 93–106.

[4] D. K. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, Y. Zhang, Optimizing cost and performance for multihoming, in: Proc. SIGCOMM, ACM, Portland, OR, USA, 2004, pp. 79–92.

[5] P. Ostovari, J. Wu, A. Khreishah, The benefits of cooperation between the cloud and private data centers for multi-rate video streaming, in: Proc. of the he 23rd International Conference on Computer Communications and Networks (ICCCN), 2014.

[6] P. Ostovari, A. Khreishah, J. Wu, Cache content placement using triangular network coding, in: Wireless Communications and Networking Conference (WCNC), IEEE, 2013, pp. 1375–1380.

[7] A. Khreishah, I. Khalil, A. Gharaibeh, H. Bany Salameh, R. Alasem, Joint caching and routing for greening computer networks with renewable energy source, in: FiCloud, 2014.

[8] A. Gharaibeh, A. Khreishah, I. Khalil, J. Wu, Asymptotically-optimal incentive-based en-route caching scheme, in: IEEE MASS, 2014.

[9] A. Khreishah, J. Chakareski, Collaborative caching for multicell-coordinated systems, in: CNTCV 2015, In conjunction with IEEE INFOCOM, 2015.

[10] J. Chakareski, Cost-performance optimization of network flow allocation and data center placement in online multi-service provider networks, in: Proc. IEEE Packet Video Workshop, IEEE, Munich, Germany, 2012, pp. 101–106.

[11] H. Xu, B. Li, Joint request mapping and response routing for geo-distributed cloud services, in: INFOCOM, 2013 Proceedings IEEE, IEEE, 2013, pp. 854–862.

[12] Z. Zhang, M. Zhang, A. G. Greenberg, Y. C. Hu, R. Mahajan, B. Christian, Optimizing cost and performance in online service provider networks., in: NSDI, 2010, pp. 33–48.

[13] H. Yuan, J. Bi, W. Tan, B. H. Li, Cawsac: Cost-aware workload scheduling and admission control for distributed cloud data centers.

[14] Z. Liu, M. Lin, A. Wierman, S. Low, L. L. H. Andrew, Greening geographical load balancing, in: Proc. SIGMETRICS, ACM, 2011.

[15] S. Huang, A. Ramamoorthy, M. Médard, Minimum cost mirror sites using network coding: Replication versus coding at the source nodes, IEEE Transactions on Information Theory 57 (2) (2011) 1080–1091.

[16] D. Leong, T. Ho, R. Cathey, Optimal content delivery with network coding, in: Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on, IEEE, 2009, pp. 414–419.

[17] A. Gandhi, M. Harchol-Balter, M. Kozuch, et al., Are sleep states effective in data centers?, in: Green Computing Conference (IGCC), 2012 International, IEEE, 2012, pp. 1–10.

[18] T. Horvath, K. Skadron, Multi-mode energy management for multi-tier server clusters, in: Proceedings of the 17th international conference on Parallel architectures and compilation techniques, ACM, 2008, pp. 270–279.

[19] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, N. McKeown, Elastictree: Saving energy in data center networks., in: NSDI, Vol. 10, 2010, pp. 249–264.

[20] Y. Shang, D. Li, M. Xu, Energy-aware routing in data center network, in: Proceedings of the first ACM SIGCOMM workshop on Green networking, ACM, 2010, pp. 1–8.

[21] IBM ILOG CPLEX optimization studio.
URL `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`

[22] N. McKeown, Software-defined networking, INFOCOM keynote talk 17 (2) (2009) 30–32.

[23] K. Kirkpatrick, Software-defined networking, Communications of the ACM 56 (9) (2013) 16–19.

[24] H. Kim, N. Feamster, Improving network management with software defined networking, Communications Magazine, IEEE 51 (2) (2013) 114–119.

[25] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, P. Rodriguez, Greening the internet with nano data centers, in: Proceedings of the 5th international conference on Emerging networking experiments and technologies, ACM, 2009, pp. 37–48.

[26] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, B. Leong, A random linear network coding approach to multicast, Information Theory, IEEE Transactions on 52 (10) (2006) 4413–4430.

[27] L. Chen, T. Ho, S. H. Low, M. Chiang, J. C. Doyle, Optimization based rate control for multicast with network coding, in: INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, IEEE, 2007, pp. 1163–1171.

[28] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge university press, 2004.

[29] D. P. Bertsekas, A. Nedi, A. E. Ozdaglar, et al., Convex analysis and optimization.

[30] D. P. Bertsekas, J. N. Tsitsiklis, Parallel and distributed computation: Numerical methods.

[31] D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, F. Zhao, Minimum-cost multicast over coded packet networks, Information Theory, IEEE Transactions on 52 (6) (2006) 2608–2623.

[32] L. Rao, X. Liu, L. Xie, W. Liu, Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment, in: INFOCOM, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–9.

[33] Google official blog.
URL `http://googleblog.blogspot.com/2009/01/powering-google-search.html`

[34] A.-H. Mohsenian-Rad, A. Leon-Garcia, Energy-information transmission tradeoff in green cloud computing, Carbon 100 (2010) 200.

[35] Google cloud platform.
URL `https://cloud.google.com/products/cloud-storage/#pricing`

[36] M. Batayneh, D. A. Schupke, M. Hoffmann, A. Kirstaedter, B. Mukherjee, On routing and transmission-range determination of multi-bit-rate signals over mixed-line-rate wdm optical networks for carrier ethernet, IEEE/ACM Transactions on Networking (TON) 19 (5) (2011) 1304–1316.

[37] V. C. Coroama, L. M. Hilty, E. Heiri, F. M. Horn, The direct energy demand of internet data flows, Journal of Industrial Ecology 17 (5) (2013) 680–688.

[38] HP, Power regulator for proliant servers.
URL `http://h20565.www2.hp.com/hpsc/doc/public/display?docId=emr{_}na-c00300430`

Table 1: Major Symbols

Symbol	Definition
$S = (S_1, \ldots, S_M)$	Number of blocks per file (M files in total)
H_k	Sending rate of the k-th file which is equal to the receiving rate
I_{ik}	$\begin{cases} 1 & \text{if node } i \text{ is requesting file } k \\ 0 & \text{otherwise} \end{cases}$
R_{ik}	Storage amount allocated to file k at node i
f_l	Total flow amount on link l
d_i	Total traffic load of server i
D_i	Total storage amount allocated at server i
X_l^{ik}	Flow amount allocated to file k destined for node i on link l
P_{ik}^j	$= \begin{cases} 1 & \text{if node } i \text{ has the } j\text{-th block for file } k \\ 0 & \text{otherwise} \end{cases}$
y_{lj}^{ik}	$\begin{cases} 1 & \text{if the } j\text{-th block for file } k \text{ destined to node } i \text{ uses link } l \\ 0 & \text{otherwise} \end{cases}$
$I_i^k(u)$	Set of previous-hop nodes for node u on the shortest path from the location of the k-th file to node i
$O_i^k(u)$	Set of next-hop nodes for node u on the shortest path from the location of the k-th file to node i
$C_1(d_i)$	Cost function of the total traffic load of server i
$C_2(D_i)$	Cost function of the total storage amount allocated at server i
$C_3(f_l)$	Cost function of the total flow amount on link l
Z_l^k	Slack variable
$\mathcal{S}$	Set of states that the demand can exhibit
$p_{ij}^{\hat{S}}$	The probability that node i will request service j given state $\hat{S}$
$Q_{ij}^{\hat{S}}$	$= \begin{cases} 1 & \text{if node } i \text{ recieved full service } j \text{ given state } \hat{S} \\ 0 & \text{otherwise} \end{cases}$
$\mathcal{T} = \{t_0, t_1, \ldots, t_{T-1}\}$	Set of time instances at which we are interested in solving the problem
$\mathcal{P}$	Stochastic matrix

Table 2: Topology Effect on the Performance of All Schemes

Scheme	US Backbone Topology	Random Topology
All File	48%	71%
Coding	35%	51%
Heuristic 1	45%	42%
Heuristic 2	86%	73%
Heuristic 3	50%	58%

Table 3: Popularity Effect on the Performance of All Schemes

Scheme	Power Set Popularity	Unary Popularity
All File	48%	34%
Coding	35%	16%
Heuristic 1	45%	32%
Heuristic 2	86%	89%
Heuristic 3	50%	34%