Content Fragmentation: A Redundancy Scheme to Save Energy in Cloud Networks

Yu Wu[1], Massimo Tornatore[1,2], Charles U. Martel[1], and Biswanath Mukherjee[1]
[1]University of California, Davis [2]Politecnico di Milano, Italy
Email: {yuuwu, mtornatore, cumartel, bmukherjee}@ucdavis.edu

Abstract**—Due to rapid growth of content-based cloud services (e.g., video streaming), energy usage of cloud infrastructures, consisting of distributed Data Centers (DCs) interconnected by a high-bandwidth long-distance network, keeps increasing. This energy usage could further increase due to requested content redundancy to ensure content-service resiliency. Typical content redundancy schemes are based on Content Replication (CR), i.e., each content is replicated in at least one secondary DC location, reachable in case of failure affecting the primary location, which induces at least 100% increase in storage energy consumption.**

To reduce storage energy overhead of CR, we investigate a new redundancy scheme, called Content Fragmentation (CF). CF exploits Reed-Solomon erasure code to fragment and encode content into blocks with less storage overhead (thus less storage energy usage) to guarantee content-service resiliency. But CF requires additional energy for content reconstruction and data transport in the core network. To determine which scheme is more energy efficient, we formulate, for both, the content-placement problem using Mixed Integer Liner Program (MILP), with the objective to minimize energy consumption. Also, due to MILP's poor scalability, we propose a Meta-heuristic Content Placement and Routing Assignment algorithm (M-CPRA) for more efficient solutions. We observe the impact on energy usage of three main metrics, i.e., number of content requests (popularity), resiliency, and latency. Results from a realistic case study suggest that CF always consumes less energy than CR given the same resiliency requirement, and CF is particularly energy-efficient when stored contents are less popular and latency-tolerant.

A part of this work was presented at IEEE ICC conference in 2016, "Green and Low-Risk Content Placement in optical content delivery networks" [6].

Index Terms—content-based cloud service, content placement, Content Fragmentation (CF), Content Replication (CR), energy optimization, resiliency, latency.

I. Introduction

Data generated and exchanged over the Internet keeps growing, and is estimated to reach 44 zettabytes by 2020 [1]. It consists typically in contents accessible to users, such as web-based (text, graphics), multi-media (video streaming), and other downloadable contents (files, software). As of 2016, 79 percent of total Internet traffic was multi-media content [1].

Contents are stored at geographically-distributed locations (e.g., DCs) and delivered to users via transport networks (e.g., Content Delivery Networks (CDN)). In pursuit of lower cost and higher Quality of Service (QoS), Internet Content Providers (ICPs) tend to switch from self-owned infrastructure to cloud (third-party compute/storage resources and network connectivity) owned by Cloud Infrastructure Providers (CIPs). For example, in early January 2016, Netflix moved its video-streaming services to Amazon Web Services (AWS) and shut down its own DCs [2]. This increases cloud energy consumption, and makes energy optimization from CIP’s perspective an important research problem.

We solve the problem of reducing the energy consumption of cloud infrastructures while guaranteeing resilient access to content services. To guarantee resiliency, content is usually replicated across multiple DCs to provide backup in case the primary content becomes unavailable or unreachable due to failures. This scheme is called Content Replication (CR) [3]. However, CR results in large content redundancy, thus increasing energy consumption for storage. Content redundancy is defined as the ratio of the total amount of storage used by a content to the content size (typically this value is 2 or 3 for CR, according to [4]). To decrease redundancy, we propose a new content redundancy scheme, called Content Fragmentation (CF), which fragments content into data blocks, and encodes data blocks into parity blocks by using Reed-Solomon erasure code [5]. Since the number of encoded parity blocks is smaller than the number of data blocks, CF redundancy is typically lower than 2 [4], and thus can help reduce energy consumption.

On the other hand, CF consumes more energy during content retrieval and reconstruction. To prevent a failure from damaging or disconnecting enough data/parity blocks to reconstruct original content, blocks should be placed in more than one DCs. Thus, CF retrieves content via multiple paths, resulting in more energy to be consumed in the core network compared to CR. Also, after enough data/parity blocks are retrieved, content reconstruction needs to be performed, causing extra energy consumption. Thus, it is important to evaluate which scheme, CF or CR, is more energy efficient.

Our study formulates, for both schemes, the content-placement problem with the goal to minimize total energy consumption. The problem is constrained by two major QoS requirements: resiliency and latency. We solve the problem optimally using MILP. Due to MILP's poor scalability, we also propose a meta-heuristic algorithm, called M-CPRA, to obtain quasi-optimal scalable solutions. Using M-CPRA solutions, we compare energy consumption of CF and CR, and evaluate the impact of three metrics: number of content requests (popularity); resiliency; and latency. Finally, we provide recommendations on which scheme to use under what conditions based on the evaluation results. Compared to our prior work [6], the new contributions are three-fold: (1) we improve the resiliency model to serve a more realistic failure scenario; (2) we propose M-CPRA meta-heuristic algorithm to overcome the scalability issue of MILP; and (3) we evaluate the two redundancy schemes, CF and CR, more thoroughly by using more metrics.

The rest of the study is organized as follows. We present an overview of related works in Section II. We introduce CF and CR schemes in Section III. In Section IV, we elaborate on how to model the two major QoS constraints (resiliency and latency). In Section V, we illustrate our energy-consumption model. Section VI presents our proposed content-placement problem together with its two solution methods: MILP and M-CPRA. Section VII discusses some illustrative numerical results. Section VIII concludes our work.

II. Related Work

We organize existing research works on cloud content placement problem into three categories based on their common minimization objectives, as shown in Table 1.

Category	Objectives	Novelties/differences of our work
Cost	Byte transfer cost [7][10-12]	1. We minimize energy consumption which is in line with cost. 2. Our CF scheme helps further to reduce energy consumption/cost.
	Storage cost [8][10-12]	
	Other operational cost [9][10-12]	
Resiliency	Risk of service disruption due to single failure [15]	1. We take both resiliency and latency into consideration. 2. We treat both of them as QoS constraints which are imposed by ICPs when signing SLA with CIPs.
	Risk of service disruption due to multiple failures [16]	
Latency	Distance [17][18]	
	Communication/access delay [18]	
	Traffic volume [18]	

Table 1 Categorized research works on cloud content placement and the differences of our work.

The first category contains studies targeting cost reduction [7-12]. Ref. [7] minimizes byte transfer cost, subject to disk space and bandwidth limit. Besides byte transfer cost, Ref. [8] also considers storage cost when placing content to achieve a balance between content storage cost and byte transfer cost. (As intuition suggests, more content replicas (higher storage cost), if distributed properly across geographically located DCs, can reduce latency, thus reducing byte transfer cost.) Ref. [9] proposes a model to minimize operational costs. Refs. [10-12] attempt to optimize the above-mentioned cost components together. Our work focuses on minimizing energy consumption (note that minimizing energy consumption does not contradict with the objective of minimizing cost, as these two objectives are positively correlated). Moreover, the major novelty in our work is the proposed CF scheme, which has potential to reduce storage usage, and hence energy consumption and cost.

A second frequently-studied objective is resiliency. Since accidental service disruptions or targeted attacks may cause significant revenue losses, researchers have developed several resiliency techniques to cope with different types of failures in cloud infrastructures [13-14]. Here, we only mention research works focusing on content-based services. Ref. [15] introduces connectivity and availability as critical resiliency metrics to measure the reachability of content after failure caused by either network disconnection or content loss, and proposes a DC/content placement problem to minimize risk from potential failures. Ref. [16] defines k-content connectivity as reachability of at least one content from any point of a DC network after multiple failures occur, and presents new strategies for flexible content placement to minimize service disruption risk.

Other research works focusing on minimizing latency fall in the third category. The objectives of latency-minimization problems include, but are not limited to, distance, communication/access delay, or traffic volume between users and DCs [17-18]. These works mostly intend to guarantee QoS.

Instead of treating resiliency and latency as objectives, our study treats them as two important QoS constraints, as they are two common metrics ICPs specify when signing Service-Level Agreements (SLAs) with CIPs.

III. Two Content Redundancy Schemes

This section introduces CF and CR schemes and how content is retrieved in both schemes.

A. Content Fragmentation (CF) Scheme

In CF, first, each given content is fragmented into k equal-size data blocks. Then, a specific type of erasure code, Reed-Solomon (RS), is used to perform content encoding, i.e., encoding k data blocks into r parity blocks of same size. Resulting data blocks and parity blocks are placed across DCs as shown in Fig. 1.

In CF, content can be retrieved using reverse-multicast, i.e., a collection of k blocks from at most k DCs must be selected, and a path from each selected DC to the user must be created. The selected k blocks could be all data blocks (Fig. 1(a)), a mix of data blocks and parity blocks (Fig. 1(b)), or all parity blocks (Fig. 1(c), noting that this scenario happens only if $k \leq r$), as RS erasure code allows any k blocks out of $(k+r)$ blocks to reconstruct the original content. Content reconstruction can be performed in a server dedicated for this purpose. Its logic is presented in Fig. 2. Upon retrieval, the number of missing data blocks (x) is counted. If x is 0 (Fig. 1(a)), the received data blocks are directly sent for re-ordering before delivering to user. Otherwise (Figs. 1(b) and 1(c)), the obtained blocks are sent into a decoder to decode the missing data blocks. The decoder contains k buffers (each for one block) and parallel decoding units to facilitate fast decoding [19]. At last, the obtained data blocks with decoded data blocks are reordered for delivery.

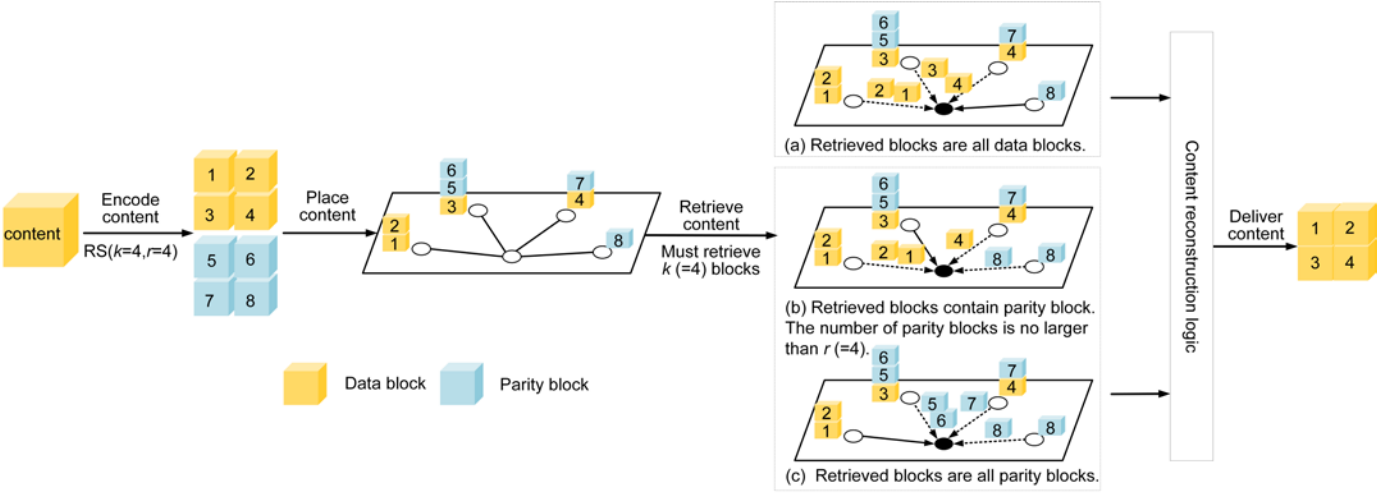

content
Encode content
RS(k=4,r=4)
Place content
Retrieve content
Must retrieve k (=4) blocks
Data block
Parity block
(a) Retrieved blocks are all data blocks.
(b) Retrieved blocks contain parity block. The number of parity blocks is no larger than r (=4).
(c) Retrieved blocks are all parity blocks.
Content reconstruction logic
Deliver content
1
2
3
4
5
6
7
8

Fig. 1 Content Fragmentation (CF) scheme.

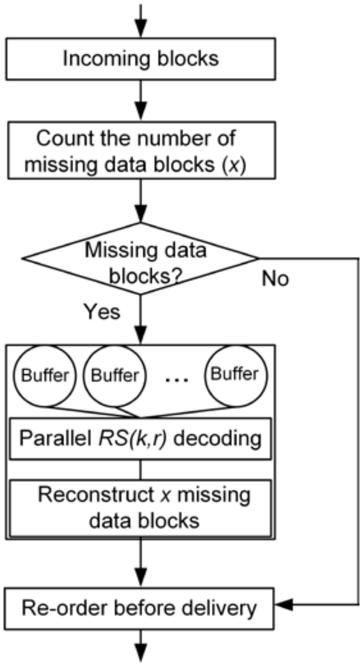

Incoming blocks
Count the number of missing data blocks (x)
Missing data blocks?
No
Yes
Buffer
Buffer
...
Buffer
Parallel RS(k,r) decoding
Reconstruct x missing data blocks
Re-order before delivery

Fig. 2 Content reconstruction logic.

B. Content Replication (CR) Scheme

CR maintains multiple replicas of the same content over geographically-distributed DCs. In CR, content can be accessed using anycast routing (i.e., a user can be served by any of the DCs hosting that content). CR improves content availability, at the expense of significant increase in storage usage.

Symbol	Meaning
Parameters	
$G(S,E)$	Network topology, including node set *S* and link set *E*
$RS(k, r)$	Reed-Solomon erasure code
DS	DC set
I	Content block index set in CF
I'	Content replica index set in CR
B	Content block size
η	Number of content replicas
N_s	Number of content requests to be served from node *s*
θ_d	Average PUE value at DC *d*
L_{th}	Latency bound (upper bound)
U_{th}	Resiliency backup level (lower bound)
H_{sd}	Number of hops from node *s* and DC *d*
D_{sd}	Distance from node *s* and DC *d*
Variables	
A_s^{id}	Binary, 1 if content requests from node *s* are served by *i*-th block/replica at DC *d*
C_{id}	Binary, 1 if block/replica *i* is placed at DC *d*
O_{sd}	Number of blocks/replicas at DC *d* serving request from node *s*
X_{sd}	Number of backup block/replica candidates for the primary blocks/replicas serving node *s* at DC *d*

Table 2 Parameters and variables.

IV. Quality of Service (QoS) Constraints

We consider two QoS constraints when solving the content-placement problem: (1) latency; and (2) resiliency. Note that variables and parameters used are listed in Table 2.

A. Latency

In general, latency experienced by a user is defined as the amount of time elapsed from the moment the user sends out a content request until the content arrives. In this study, we assume that networks and DCs have enough capacity so that processing delay and queueing delay are kept under budget, and not modeled. Our focus is on content propagation delay for both schemes. Thus, the latency in CF is defined as Round-Trip (propagation) Time (RTT) from the user to its most-distant serving DC. Note that, in CF, if multiple blocks are retrieved from a single DC location or through the same link/route, we assume that they are retrieved simultaneously (e.g., via multiple wavelengths in Wavelength-Division Multiplexing (WDM) networks). Thus, only delay for one block is counted. The latency in CR can be defined as RTT from the user to its serving DC.

We use Latency Island (LI) to model latency for both CF and CR. Given a pre-defined latency bound (L_{th}), a node's LI is defined as a set of nodes which can be reached by the current node via a path[1] no longer than L_{th}, shown as follows:

$$LI(s) = \{d \in DS \mid dis(d,s) \leq L_{th}\} \ \forall s \in S \qquad (1)$$

where S and DS are the topology node set and DC set, respectively. With this definition, constraining latency for each node in the network is equivalent to guaranteeing that its primary content must be placed within its LI. For CF, the latency constraint is translated into the following equation:

$$\sum_{d \in LI(s)} \sum_{i \in I} A_s^{id} = k \ \forall s \in S \qquad (2)$$

It means that the primary content to be placed within each node's LI consists of k content blocks. A_s^{id} is a binary variable indicating whether or not service requests from node s are served by i-th block at DC d. Set I is the block index set and is formulated as follows:

$$I = \{i \in Z \mid 1 \leq i \leq k + r\} \qquad (3)$$

For CR, the latency constraint is formulated as follows:

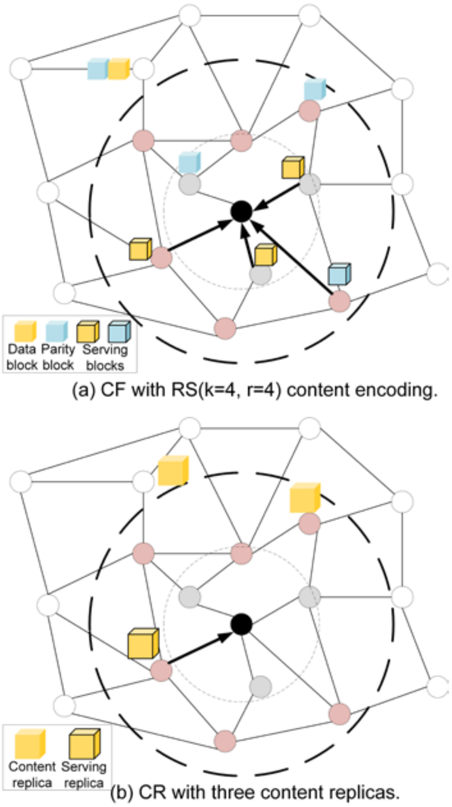
$$\sum_{d \in LI(s)} \sum_{i \in I'} A_s^{id} = 1 \ \forall s \in S \qquad (4)$$

It means that a single content replica, serving as primary content, must be placed within each node's LI. A_s^{id} is a binary variable indicating whether or not service requests from node s are served by i-th replica at DC d. I' is the replica index set and is formulated as follows:

$$I' = \{i \in Z \mid 1 \leq i \leq \eta\} \qquad (5)$$

Fig. 3 presents examples for both schemes. We assume the black node is retrieving a content. The two circles represent two LIs of the black node (at the center of both circles) with different latency bounds. Grey nodes are reachable from the black node within small LI, and, together with pink nodes, are reachable from black node within large LI. In (a) for CF, although three content blocks are placed within small LI, they are

[1] We use shortest path between a node pair for potential content delivery, as it results in largest energy reduction if core network capacity is not the bottleneck. In general, k-shortest paths between each node pair are often used for flexible routing [15]. The path to be used while satisfying latency constraint can be pre-configured for different purposes such as energy saving, failure avoidance, etc.



(a) CF with RS(k=4, r=4) content encoding.

(b) CR with three content replicas.

Fig. 3 Latency constraint examples.

not enough to reconstruct the content, as at least four blocks are needed in RS(*k*=4, *r*=4). Thus, the latency constraint for the black node is not satisfied. In (b) for CR, small LI does not include any node with a content replica placed at it. Thus, the latency constraint for the black node cannot be satisfied either. But, for the larger LI, six content blocks and two content replicas are incorporated for CF in (a) and CR in (b), respectively. CF can by served by any four out of six blocks while CR can be served by any one out of two replicas. Thus, latency constraint is satisfied in both schemes.

B. *Resiliency*

In this study, we consider a single-failure scenario, in which only one failure may occur at one DC location. To prevent such failure from disrupting service, a backup content which is not currently serving a user must be reachable in case the primary content becomes unavailable. The backup content needs to be: (1) node-disjoint with the primary content, and (2) within the user's LI to guarantee latency constraint.

We formally define resiliency constraint for both schemes. For CF, we first obtain the number of available backup block candidates that can be used to backup the primary content blocks located at node *d* serving node *s* (X_{sd}), as shown below:

$$X_{sd} = \sum_{d^* \in LI(s)\backslash d} \sum_{i \in I} C_{id^*} - \sum_{d^* \in LI(s)\backslash d} \sum_{i \in I} A_s^{id^*} \ \forall s \in S \ d \in DS \qquad (6)$$

It is calculated by subtracting the number of content blocks that are serving node *s* located within node *s*'s LI but not at node *d* from the total number of content blocks that are located within node *s*'s LI but not at node *d*. C_{id^*} is a binary variable indicating whether or not *i*-th block is placed at DC d^*. Then, we obtain the number of primary content blocks at node *d* serving node *s* (O_{sd}) as follows:

$$O_{sd} = \sum_{i \in I} A_s^{id} \ \forall s \in S \ d \in DS \qquad (7)$$

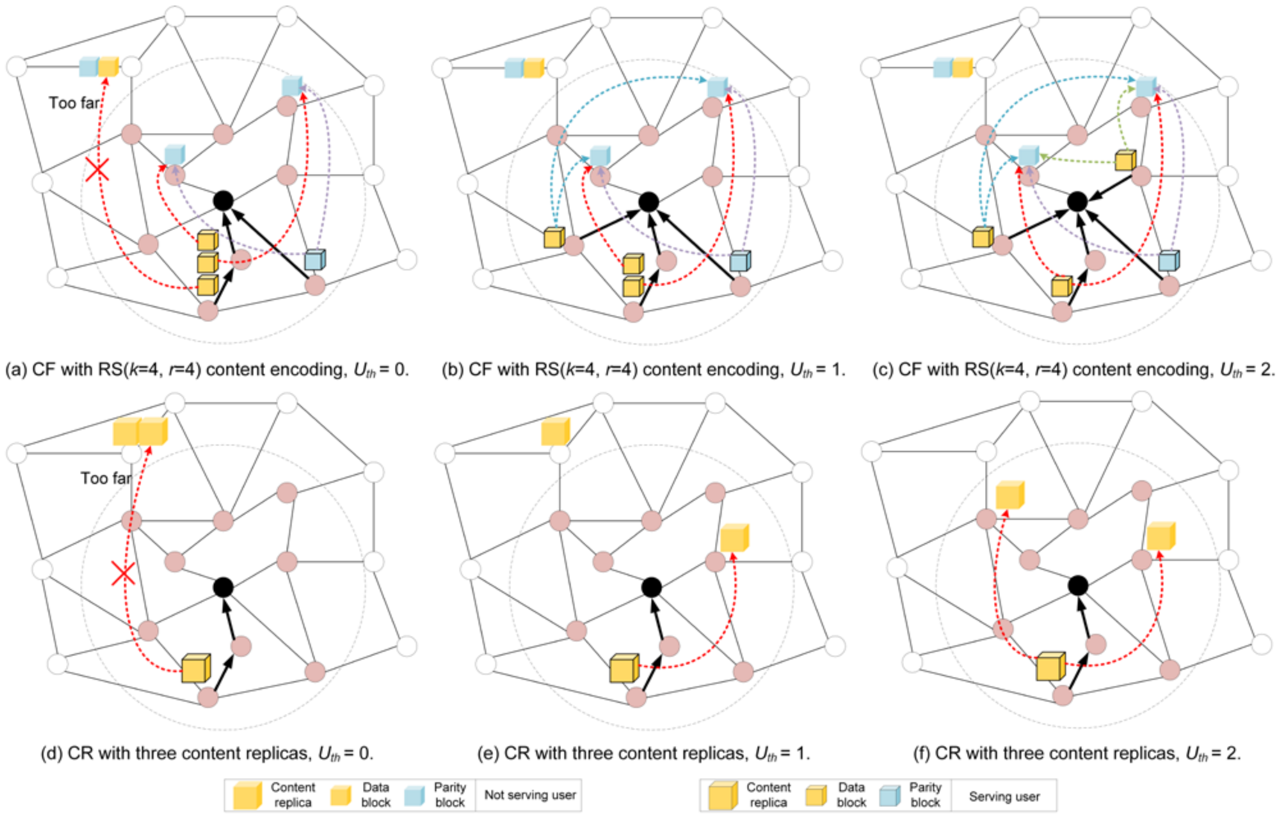
It is calculated by adding up all content blocks serving node *s* at node *d*. We use a pre-determined value, named resiliency backup level (U_{th}), to measure the redundancy of the backup blocks. Thus, the resiliency constraint is given by:

$$X_{sd} \geq U_{th} \cdot O_{sd} \ \forall s \in S \ d \in DS \qquad (8)$$

It means that node-disjoint backup block candidates for node *d* must outnumber blocks at node *d* serving node *s* by a factor of U_{th} in order to provide required backup.

The resiliency formulation for CF in Eqns. (6)-(8) can be adapted to CR by replacing *I* by I' in Eqns. (6)-(7). In CR, U_{th} in Eqn. (8) now measures the redundancy of the backup replicas.

Fig. 4 presents examples for both schemes. Again, we assume the black node whose LI includes the nodes colored in pink is retrieving content. Content placement, primary content blocks/replicas (with the emboldened outline) serving the black node, and content-retrieval paths (emboldened arrows) are shown together. Dashed arrows point to possible backup contents. For CF, if there are multiple primary blocks placed at the same location as shown in (a), it is hard to find enough node-disjoint backup blocks within the same LI. Thus, single-failure resiliency is not guaranteed (corresponding to a loose resiliency backup level



(a) CF with RS(k=4, r=4) content encoding, U_{th} = 0. (b) CF with RS(k=4, r=4) content encoding, U_{th} = 1. (c) CF with RS(k=4, r=4) content encoding, U_{th} = 2.

(d) CR with three content replicas, U_{th} = 0. (e) CR with three content replicas, U_{th} = 1. (f) CR with three content replicas, U_{th} = 2.

Fig. 4 Resiliency constraint examples.

0). As the backup level increases from 0 to 2 (from (a) to (c)), primary blocks tend to be placed evenly within LI to guarantee that enough node-disjoint backup blocks are available for each of them. For CR, if there is only one content replica in LI (serving as primary replica as in (d)), there is no backup for it without compromising latency constraint (corresponding to a loose resiliency backup level 0). As the backup level increases from 0 to 2 (from (d) to (f)), the number of content replicas placed in the LI must increase to provide enough backup replicas for the primary one.

Another takeaway from Fig. 4 is the relationship between resiliency and resource usage. For CF, as we focus on single encoding scheme, increasing resiliency (U_{th}) does not increase resource usage in terms of storage. It only decentralizes the content placement. But for CR, higher resiliency (U_{th}) directly translates to more resource usage in terms of more backup replicas within reach.

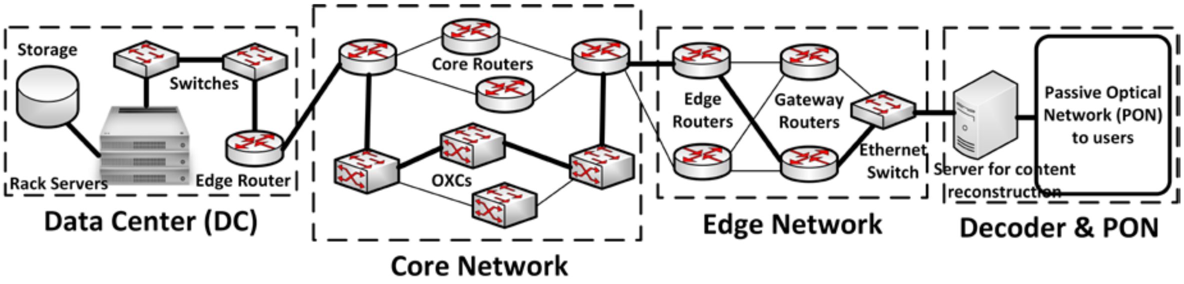


Fig. 5 Content delivery path from DC to user, adapted from [20].

Type	Energy Density	Value
E_{OXC}	Optical Crossconnect (OXC)	1.95×10^{-8} J/bit
E_l	WDM link	9.87×10^{-13} J/bit/km
E_{cr}	core router	1.2×10^{-8} J/bit
E_e	Ethernet switch	8.21×10^{-9} J/bit
E_g	gateway router	1.38×10^{-7} J/bit
E_{er}	edge router	2.63×10^{-8} J/bit
E_{sr}	server	2.81×10^{-7} J/bit
E_{st}	Storage (monthly)	2.03×10^{-4} J/bit

Table 3 Energy density of different devices [20].

V. Energy Model

Now, we model the energy consumption of cloud-content service, which will be used as the objective function, for both CF and CR schemes.

When retrieving content from a DC, we model three energy-consumption components: (1) in DCs, (2) in core network, and (3) for content reconstruction (only applied to CF). We do not model energy consumption for edge network and access network (e.g., Passive Optical Network (PON))[1]. Fig. 5 shows a content-delivery path (in bold) from DC to user. On this path, each device's energy density is shown in Table 3. The values in Table 3 are based on a broadly-adopted assumption from [21] that energy consumption is proportional to traffic and storage volume.

A. *DC Energy Consumption*

As illustrated in Fig. 5, for CF, DC energy consumption can be formulated as:

$$E_{DC} = \sum_{d \in DS} \theta_d B \left[\sum_{s \in S} N_s O_{sd} \left(2E_e + E_{er} + E_{sr} \right) + \sum_{i \in I} C_{id} E_{st} \right] \tag{9}$$

The first sum within the square bracket represents the energy consumption of Ethernet switches (E_e), edge routers (E_{er}), and servers (E_{sr}) scaled by number of content requests N_s while the second sum within the

[1] For each content request, there is always same amount of retrieved data (k blocks in CF and one replica in CR) going through edge network and access network in both schemes, leaving no energy reduction margin. Moreover, since PON is mostly passive, its energy consumption is assumed negligible.

square bracket represents the energy consumption of storage (E_{st}). Within the outer sum, B is block size. θ_d, known as Power Usage Effectiveness (PUE) for each DC d, is used to incorporate the energy consumption of non-IT equipment, such as cooling/lighting equipment, Uninterruptable Power Supply (UPS) devices, etc. [22]

Eqn. (9) can be adapted to calculate DC energy consumption for CR (E'_{DC}) by replacing B and I with kB and I', respectively, as shown below:

$$E'_{DC} = \sum_{d\in DS} \theta_d kB \left[\sum_{s\in S} N_s O_{sd} \left(2E_e + E_{er} + E_{sr}\right) + \sum_{i\in I'} C_{id} E_{st} \right] \tag{10}$$

B. Core-Network Energy Consumption

We assume the core network to be optical-bypass-enabled [23]. For CF, core network energy consumption can be formulated as follows:

$$E_C = \sum_{s\in S} \sum_{d\in DS} N_s O_{sd} B \left[2E_{cr} + E_{oxc}(H_{sd} + 1) + E_l D_{sd} \right] \tag{11}$$

The items within the square bracket represent the energy consumption of ingress/egress core routers (E_{cr}), Optical Crossconnects (E_{oxc}) scaled by number of hops along the content-delivery path (H_{sd}), and WDM links (E_l) scaled by the distance of the content-delivery path (D_{sd}), respectively. They are all scaled by number of content requests N_s.

To get core network energy consumption for CR (E'_C), B in Eqn. (11) needs to be replaced by kB, as shown follows:

$$E'_C = \sum_{s\in S} \sum_{d\in DS} N_s O_{sd} kB \left[2E_{cr} + E_{oxc}(H_{sd} + 1) + E_l D_{sd} \right] \tag{12}$$

C. Content Reconstruction Energy Consumption

Content reconstruction only applies to CF. Its energy consumption depends on number of missing data blocks to be decoded, i.e., number of parity blocks obtained during content retrieval for each content request. Content reconstruction energy consumption can be formulated as follows:

$$E_R = \sum_{s\in S}\sum_{d\in DS}\sum_{i=k+1}^{k+r} A_s^{id} E_{sr} N_s B \tag{13}$$

where the sum indexed by *i* within range [*k+1*, *k+r*], together with binary variable A_s^{id} are used to count the number of obtained parity blocks for each content request.

D. *Total Optimizable Energy*

Putting together energy consumption of different network components, total optimizable energy consumption of content service for CF (*ξ*) and CR (*ξ'*) can be expressed as follows:

$$\xi^{(')} = E_{DC}^{(')} + E_C^{(')} + E_R \tag{14}$$

VI. PROBLEM FORMULATION AND SOLUTION METHODS

Now, we state the content-placement problem and two solution methods, MILP and M-CPRA, for both CF and CR.

A. *Problem Statement*

Given network topology *G(S, E)*, DC locations (DS, $DS \subseteq S$), equipment energy values (Table 3), and content requests (N_s), we solve the content blocks/replicas-placement problem for both CF and CR under latency and resiliency constraints. The goal is to minimize total energy consumption as described in Section V.

B. *MILP Formulation*

The objective function is:

$$minimize\ \xi^{(')} \tag{15}$$

subject to the following constraints:

$$\sum_{d\in DS} C_{id} = 1 \quad \forall i \in I^{(')} \tag{16}$$

Eqn. (16) states that each block/replica in CF/CR has to be placed at one DC.

$$A_s^{id} \le C_{id} \quad \forall i \in I^{(')}, d \in DS, s \in S \tag{17}$$

Eqn. (17) states that node s is able to access block/replica i from DC d only if block/replica i is placed at DC d for CF/CR.

Also, we consider QoS constraints, i.e., latency and resiliency as discussed in Section IV, including Eqns. (1)-(3), (6)-(8) for CF, and Eqns. (1), (4)-(5), (6)-(8) for CR, respectively. Note that for CR, we use the smallest possible η in Eqn. (5) (obtained by iterating from 1 upwards until landing on a value at which Eqn. (15) is able to find a solution) which satisfies both QoS constraints to guarantee low resource usage for ICP.

In the MILP formulation, the number of variables is $|I^{(\cdot)}||S||DS| + |I^{(\cdot)}||DS| + 2|S||DS|$, $O(|I^{(\cdot)}||S|^2)$ ($DS \subseteq S$), and the number of constraints is $|S| + 3|S||DS| + |I^{(\cdot)}| + |I^{(\cdot)}||S||DS|$, $O(|I^{(\cdot)}||S|^2)$. As number of content blocks/replicas and size of network become large, the problem size can grow significantly towards limited scalability.

C. Meta-heuristic: Content Placement and Routing Assignment (M-CPRA) Algorithm

Since MILP is not scalable, we also propose a meta-heuristic algorithm, called M-CPRA. M-CPRA moves towards optimal solution fast with much lower computational complexity than MILP. The flow chart of M-CPRA is shown in Fig. 6. It is applicable for both CF and CR. At first, it randomly initializes a placement. Next, M-CPRA can be roughly described in three steps: (1) constructing neighborhood (which is a collection of solutions obtained from the local optimal solution of the last neighborhood), (2) escaping from the local optimal solution, and (3) approaching the optimal solution.

When constructing the neighborhood, M-CPRA iteratively generates γ new placements by probabilistically rearranging the placement of the local optimal solution obtained from the last neighborhood. Pseudo code in Fig. 7 shows an algorithm named PRP to generate a new placement. We summarize it in three steps. First, a random number $\omega^{(\cdot)}$ is generated denoting the number of blocks/replicas to be rearranged. Second, $\omega^{(\cdot)}$ blocks/replicas are removed sequentially from DCs. The removal process follows a probabilistic distribution $Pr^R(d)$ as defined in Eqn. (18) which represents the probability of DC d being selected to remove a content block/replica. $Pr^R(d)$ depends on two factors: (1) number of LIs DC d belongs to (δ_d) as in the first component; and (2) DC d's energy efficiency in terms of its PUE value (θ_d) as in the second component. In the first

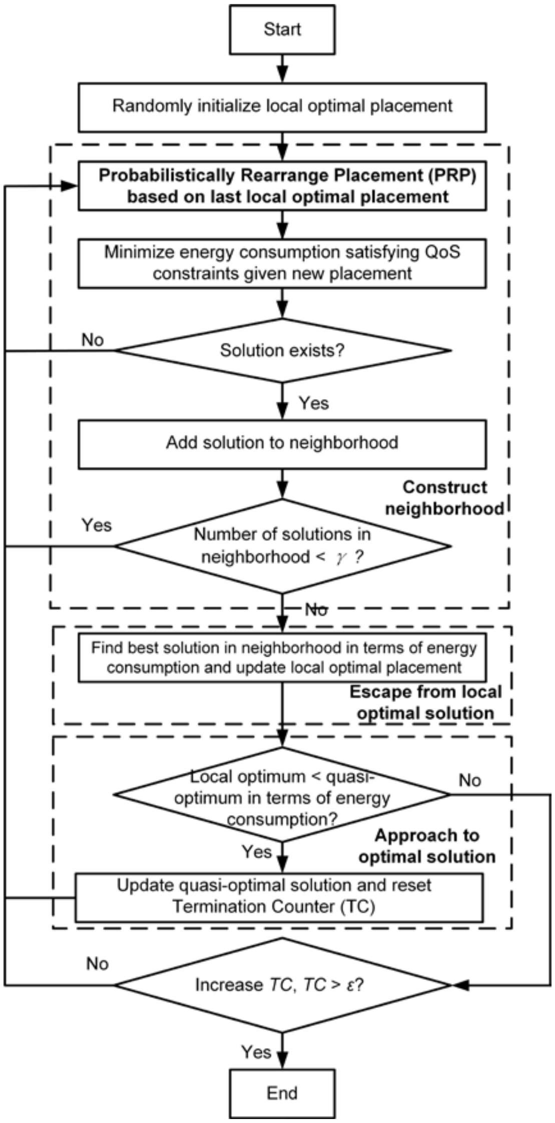

Start
Randomly initialize local optimal placement
Probabilistically Rearrange Placement (PRP) based on last local optimal placement
Minimize energy consumption satisfying QoS constraints given new placement
No
Solution exists?
Yes
Add solution to neighborhood
Construct neighborhood
Yes
Number of solutions in neighborhood < γ ?
No
Find best solution in neighborhood in terms of energy consumption and update local optimal placement
Escape from local optimal solution
Local optimum < quasi-optimum in terms of energy consumption?
No
Yes
Approach to optimal solution
Update quasi-optimal solution and reset Termination Counter (TC)
No
Increase TC, TC > ε?
Yes
End

Fig. 6 M-CPRA algorithm flow chart.

```
Probabilistically Rearrange Placement (PRP)
Input: Local optimal placement from last neighborhood (block/replica-hosting DC set ($HS$) + set of blocks/replicas placed at DC $d$ in $HS$ ($\phi_d$));
Output: Resulting placement ($HS^* + \phi^*_d$ (equal $HS$ and $\phi_d$ initially)).
Randomly generate number of blocks/replicas to be rearranged ($\omega^{(\cdot)} \leq |I^{(\cdot)}|$)
repeat
    Count number of LIs each DC $d$ belongs to in $HS^*$ ($\delta_d$).
    Calculate probability of DC $d$ in $HS^*$ being selected to remove a content block/replica. $0 \leq k_1, k_2 \leq 1, k_1 + k_2 = 1$.
        $\mathrm{Pr}^R(d) = k_1 \cdot \frac{1/\delta_d}{\sum_{j \in HS^*} 1/\delta_j} + k_2 \cdot \frac{\theta_d}{\sum_{j \in HS^*} \theta_j} \quad \forall d \in HS^*$    (18)
    Probabilistically select a DC $d$ from $HS^*$ according to the above distribution, remove the block/replica with smallest index number from it, and update $HS^*$ and $\phi^*_d$ if necessary.
until $\omega^{(\cdot)}$ blocks/replicas get removed
Use Eqn. (18) to calculate $\mathrm{Pr}^R(d)$ for each DC $d$ in DC set $DS$.
Calculate probability of DC $d$ in $DS$ being selected to add a block/replica.
        $\mathrm{Pr}^A(d) = \frac{1 - \mathrm{Pr}^R(d)}{|DS| - 1} \quad \forall d \in DS$    (19)
for each removed block/replica
    Probabilistically select a DC $d$ from $DS$ according to the distribution in Eqn. (19), add the removed block/replica to it, and update $HS^*$ and $\phi^*_d$ if necessary.
end
```

Fig. 7 PRP pseudo code.

component, the reciprocal of δ_d is taken to guarantee that DC d is more likely to be selected to remove a content block/replica if it belongs to fewer LIs. Second component means that the higher DC d's PUE value is, the more likely DC d will be selected to remove a content block/replica. Coefficients k_1 and k_2 ($0 \leq k_1, k_2 \leq 1, k_1 + k_2 = 1$) are used: (1) to scale the sum of all $Pr^R(d)$'s to 1; and (2) to assign weights to two components. Third, each removed content block/replica is added back based on probabilistic distribution $Pr^A(d)$ defined in Eqn. (19). $Pr^A(d)$ is the opposite of $Pr^R(d)$ ($1-Pr^R(d)$). It represents the probability of DC d being selected to add the removed content block/replica back. The denominator ($|DS|-1$) equals the sum of all ($1-Pr^R(d)$)'s to guarantee that the sum of $Pr^A(d)$'s equals 1.

After a new placement is generated, M-CPRA takes it as input and invokes MILP sub-routine Eqn. (15) under its constraints mentioned in Section VI.B to find a solution of energy minimization. If such a solution exists, the generated placement together with the solution is added into the neighborhood. Otherwise, M-CPRA discards current placement, as two QoS constraints are not satisfied.

The procedures mentioned in the above two paragraphs repeat until neighborhood construction is complete. Then, the placement with minimal energy consumption is selected as the new local optimal placement, which will be used to construct the next neighborhood. Doing so facilitates escaping from local optimum when optimization gets stranded.

The algorithm's best solution so far (referred to as the quasi-optimal solution) is updated if its energy consumption is larger than that of new local optimal solution. This enables M-CPRA to approach the optimal solution.

Eventually, M-CPRA terminates if its quasi-optimal solution has not been updated for more than ε times (ε is defined as termination threshold). A Termination Counter (TC) is maintained for this purpose.

In M-CPRA, time complexity for PRP algorithm is $O(|I^{(')}||S|)$. In the MILP sub-routine after PRP, since the content placement is given, the problem size (in terms of number of variables and number of constraints) reduces to $O(|I^{(')}||S|)$, much smaller than the original MILP formulation ($O(|I^{(')}||S|^2)$). Because time complexity of MILP sub-routine (exponential with problem size, denoted as $f(|I^{(')}||S|)$) dominates PRP, the overall time complexity of M-CPRA is thus $O(\varepsilon\, \gamma\, f(|I^{(')}||S|))$ as opposed to $O(f(|I^{(')}||S|^2))$ in original MILP formulation.

VII. Illustrative Numerical Examples

In this section, we present: (A) simulation settings, (B) a performance comparison between M-CPRA and MILP, and (C) numerical results of M-CPRA for CF and CR.

A. Simulation Settings

We use industry-standard content redundancy for CF (RS(10, 4) erasure code) [4]. As for CR, the number of content replicas in use is as small as possible to satisfy the latency and resiliency constraints. We only

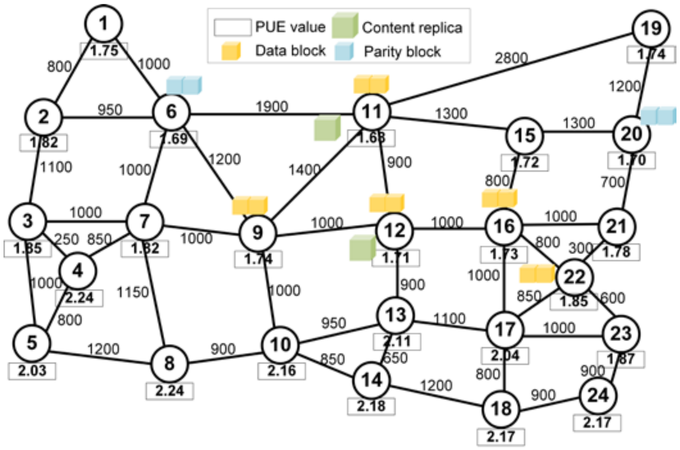


Fig. 8 USNET topology.

consider a single type of content, as our study intends to focus on the impact CF and CR have on energy consumption (in practice, it can be easily adapted to multiple-content-type scenarios). Content size is assumed to be 1 GB, and block size is 0.1 GB. Content delivery rate is assumed to be 10 Gbps. Number of total content requests varies for different simulation scenarios. For a given number of content requests, the simulation result in terms of energy consumption is averaged over 10 runs. Each simulation run is assumed to randomly distribute content requests among nodes.

We use the USNET topology shown in Fig. 8, and assume that each node is connected to a DC (DS = S). We obtain PUE value for each DC location from Ref. [24].

For our M-CPRA algorithm, we set neighborhood size (γ) to be 10, and make the two components in Eqn. (18) equally important by assigning both coefficients (k_1 and k_2) to be 0.5.

B. M-CPRA Algorithm Performance Analysis Against MILP

Given a fixed total number of content requests (N_{cr}) (N_{cr}=6,000), and a realistic QoS setting, i.e., resiliency constraint (U_{th}=1) and latency constraint (L_{th}=50 ms), we run the proposed MILP to obtain optimal solution in terms of energy consumption. To verify M-CPRA's performance, we increase termination threshold (ε) in steps of 50. Due to M-CPRA's non-deterministic characteristic, at each ε, average energy consumption ($\bar{E}$) and average run time ($\bar{T}$) across 10 runs are obtained to compare with MILP solution (E_{op} and T_{op}).

We use Figs. 9 and 10 to compare M-CPRA's performance against MILP. We focus on two metrics, i.e., M-CPRA error in Fig. 9 and relative M-CPRA run time against MILP run time (short form "M-CPRI run

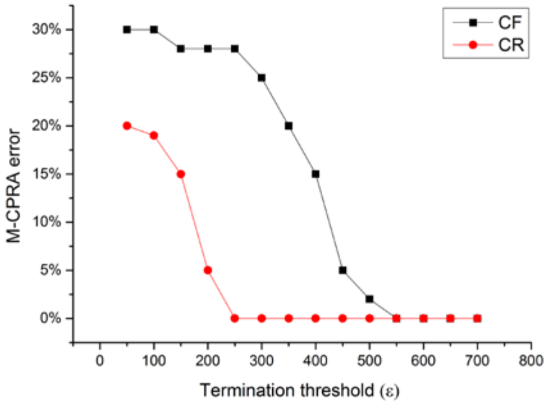


Fig. 9 M-CPRA error against termination threshold.

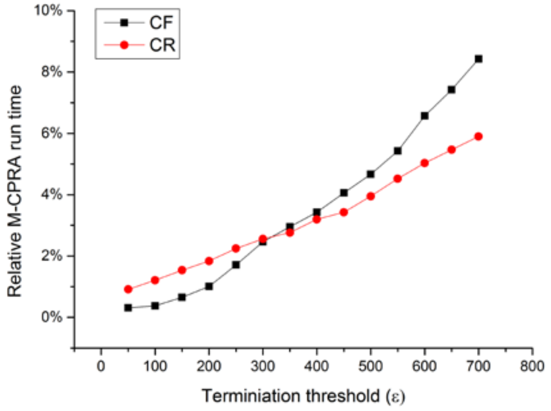
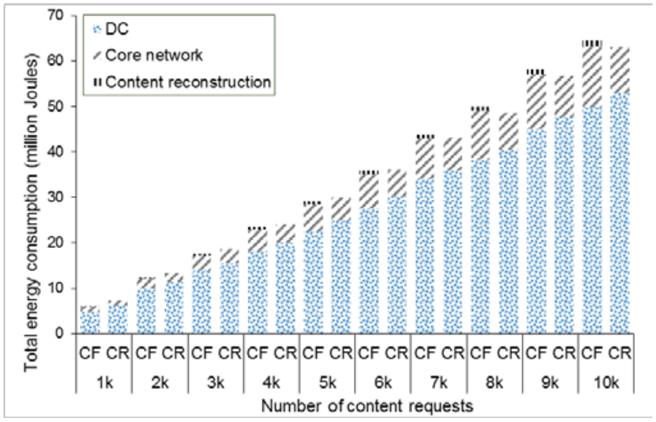


Fig. 10 Relative M-CPRA run time against termination threshold.

time" is used hereafter) in Fig. 10. M-CPRA error is defined as $(\bar{E} - E_{op})/E_{op}$ while M-CPRA run time is defined as $\bar{T}/T_{op}$. We observe from Fig. 9 that M-CPRA errors for both CF and CR decrease as *ε* increase. This is because large *ε* allows M-CPRA algorithm to search through more solutions before termination. Therefore, the final solution M-CPRA lands on is closer to the optimal solution. CF has a larger initial M-CPRA error (30.1%) as opposed to CR (20.3%), and approaches to optimal solution slower than CR (*ε* = 550 for CF when approaching to optimal solution vs. *ε* = 250 for CR). This is because CF has larger search space than CR (hence more room for energy reduction). In Fig. 10, we observe that M-CPRA run times for both CF and CR increase with the increase of *ε*. The run times for CF at *ε* = 550 and for CR at *ε* = 250 (i.e., the *ε* values at which we obtain quasi-optimal solution) where M-CPRA errors approach to 0 are 5.4% (around 5 mins) and 2.2% (around 1 min), respectively, showing that M-CPRA works well at reasonably-low run times. As *ε* increases, relative-CPRA runtime for CR grows slower than CF, confirming again the fact that the search space of CF is larger than that of CR. Also, content placements obtained for CF at *ε* = 550 and for CR at *ε* = 250 from M-CPRA are the same as those obtained from MILP, as shown in Fig. 8.

C. CF vs. CR Analysis

Now, we compare CF and CR in terms of three metrics: (1) number of content requests (N_{cr}); (2) resiliency; and (3) latency, using M-CPRA algorithm. Each simulation run is performed at *ε* = 550 for CF and *ε* = 250 for CR, respectively.



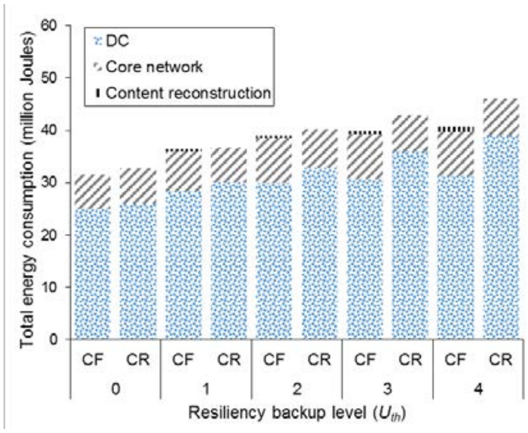


Fig. 11 Energy consumption against number of content requests.

Fig. 12 Energy consumption against resiliency backup level.

1) *Number of Content Requests*

We use the same example of Section VII.B to demonstrate the impact of changing N_{cr} on energy consumption for the two schemes. Fig. 11 shows the relation between total energy consumption (*y* axis) and N_{cr} (*x* axis). First, we observe that total energy consumption of both CF and CR increases with N_{cr} and that DCs consume a larger amount of energy compared to core network and content reconstruction (only for CF). Second, and less intuitively, CF saves energy compared to CR only when N_{cr} is small. Energy saving achieves its maximum, 17.8%, at N_{cr} = 1000, and then decreases as N_{cr} increases. Beyond N_{cr} = 6000, CF starts consuming more energy than CR. The reason CF consumes more energy than CR for large number of content requests is that some blocks are forced to be placed at locations with low PUEs up north. This increases energy consumption of multi-path routing for CF. And together with increasing energy consumption of content reconstruction, CF’s energy overhead outpaces its energy saving from DC storage.

2) *Resiliency*

To study the impact of resiliency on energy consumption, we set N_{cr} = 6,000 and latency bound (L_{th}) to be large enough (200 ms) to decouple the effect of latency on resiliency. We plot total energy consumption vs. resiliency backup level (U_{th}) in Fig. 12. We observe that, for both CF and CR, total energy consumption increases as U_{th} increases. For CR, this is because more content replicas will be needed, resulting in more storage energy consumption. For CF, this is because content blocks need to be placed in a more distributed

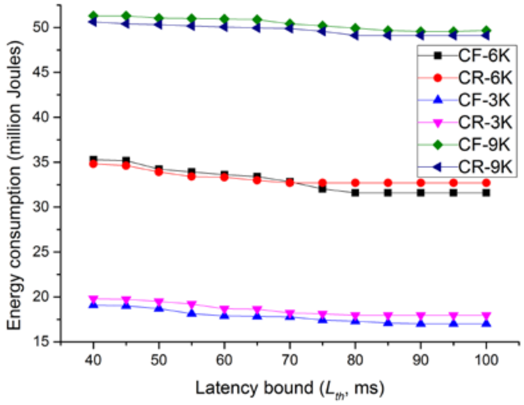


Fig. 13 Energy consumption against latency.

way, thus more likely to be placed at DC locations with high PUE values. However, as increasing U_{th} distributes content block placement, there are also a growing number of blocks tending to be placed at low-PUE DC locations. This offsets the growth rate of CF's total energy consumption, and flattens the trend after U_{th} exceeds 2. Energy consumption of CR increases faster and is always larger than that of CF, as increasing storage overhead has much more contribution to total energy consumption than placing some content blocks at high-PUE DC locations. For example, when CF reaches its maximum backup level (4 in RS(k=10, r=4), i.e., when every primary block is placed at a different location and every backup block is node-disjoint with primary blocks), the number of replicas CR requires is 5, resulting in a large storage overhead and 12.9% more total energy consumption than CF.

3) *Latency*

To study the impact that latency has on energy consumption, we set N_{cr} = 6,000 and the resiliency backup level (U_{th}) to be 0 to decouple the effect of resiliency on latency. Fig. 13 shows the result of CF and CR energy consumption for increasing values of latency. We observe that energy consumption for both CF and CR decreases when latency bound increases. This is because relaxed latency bound allows users to be served by the content blocks/replicas that are placed at distant DCs with lower PUEs. Also, in CR, more relaxed latency requirement reduces the number of content replicas required to serve users. We also observe than CF can save additional energy with respect to CR, at most 3.4% when latency bound is large (i.e., > 70 ms in our example). This is because blocks can be distributed across multiple DCs with low PUEs to guarantee energy reduction on both DCs and routing. CR can save energy when latency bound is stringent, as blocks in CF are likely to be centralized at DC locations with high PUEs.

To observe the impact of content popularity on latency requirement, we also plot the comparison of CF and CR when content is less popular (with 3K content requests in total) and when content is more popular (with 9K content requests in total) in Fig. 13. We observe that CF always saves energy compared to CR with less popular content regardless of latency, and that CR always saves energy compared to CF with more popular content regardless of latency. This confirms again the result we obtained from Fig. 11.

VIII. Conclusion

In this study, we explored the optimization of energy consumption of cloud content-based services by proposing an inter-DC content redundancy scheme named Content Fragmentation (CF). Together with an MILP solution, we proposed a meta-heuristic algorithm, called M-CPRA, to efficiently solve the content placement problem. Simulations results suggest that CF saves energy compared to CR while guaranteeing same level of resiliency. We should use CF when: (1) the number of content requests is small, and (2) latency requirement is not stringent. Considering that we only used single CF encoding scheme throughout the study, potential future work could focus on exploring the impact that different CF encoding schemes have on energy consumption.

Acknowledgement

We thank the anonymous reviewers for their constructive comments in improving the paper. This work was supported in part by NSF Grant No. CNS-1217978.

References

[1] M. Wang, *et al.*, “An overview of Cloud based Content Delivery Networks: Research Dimensions and state-of-the-art,” *Springer Transactions on Large-Scale Data-and Knowledge-Centered Systems XX*, vol. 9070, pp. 131-158, 2015.

[2] D. Chernicoff, “Netflix closes data centers and goes to public cloud,” *Datacenter Dynamics*, 2015. [Online]. Available: http://www.datacenter dynamics.com/content-tracks/colo-cloud/netflix-closes-data-centers-and-goes-to-public-cloud/94615.fullarticle.

[3] M. F. Habib, *et al.*, “Design of disaster-resilient optical datacenter networks,” *IEEE Journal of Lightwave Technology*, vol. 30, no. 16, pp. 2563-2573, 2012.

[4] W. Wang, “Saving capacity with HDFS RAID,” *Facebook Code,* 2014.

[5] S. Lesavich, *et al.*, “Method and system for electronic content storage and retrieval with galois fields on cloud computing networks,” *U.S. Patent No. 9,037,564*, 2015.

[6] Y. Wu, *et al*., “Green and Low-Risk Content Placement in optical content delivery networks,” *Proc. IEEE ICC*, 2016.

[7] D. Applegate, *et al*., “Optimal content placement for a large-scale VoD system,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2114-2127, 2016.

[8] Y. Jin, *et al*., “Toward cost-efficient content placement in media cloud: modeling and analysis,” *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 807-819, 2016.

[9] K. Katsalis, *et al*., “A cloud-based content replication framework over multi-domain environments,” *Proc. IEEE ICC*, 2014.

[10] C. Papagianni, *et al*., “A cloud-oriented content delivery network paradigm: Modeling and assessment,” *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 5, pp. 287-300, 2013.

[11] M. Hu, *et al*., “Practical resource provisioning and caching with dynamic resilience for cloud-based content distribution networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2169-2179, 2014.

[12] F. Wang, *et al*., “Migration towards cloud-assisted live media streaming,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 272-282, 2016.

[13] F. Dikbiyik, *et al*., “Minimizing the Risk From Disaster Failures in Optical Backbone Networks,” *IEEE/OSA Journal of Lightwave Technology*, vol. 32, no. 18, pp. 3175-3183, 2014.

[14] C. Colman-Meixner, *et al*., “A survey on resiliency techniques in cloud computing infrastructures and applications,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2244-2281, 2016.

[15] S. Ferdousi, *et al*., “Disaster-Aware Datacenter Placement and Dynamic Content Management in Cloud Networks,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 7, pp 681-694, 2015.

[16] X. Li, *et al*., “Content placement with maximum number of end-to-content paths in K-node (edge) content connected optical datacenter networks,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 1, pp 53-66, 2017.

[17] X. Guan, *et al*., “Push or pull? Toward optimal content delivery using cloud storage,” *Journal of Network and Computer Applications*, vol. 40, pp. 234-243, 2014.

[18] M. A. Salahuddin, *et al*., “A Survey on Content Placement Algorithms for Cloud-based Content Delivery Networks,” *IEEE Access*, 2017.

[19] K. V. Rashmi, *et al*., “A hitchhiker's guide to fast and efficient data reconstruction in erasure-coded data centers,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 331-342, 2015.

[20] K. Guan, *et al*., “On the energy efficiency of content delivery architectures,” *Proc. IEEE ICC*, 2011.

[21] P. Ruiu, *et al*., “On the Energy-Proportionality of Data Center Networks,” *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 197-210, 2017.

[22] V. Avelar, *et al*., “PUETM: A comprehensive examination of the metric,” *Green Grid White Paper #49*, 2012.

[23] J. M. Simmons, *Optical Network Design and Planning*, 2nd Edition, Springer, 2014.

[24] Y. Wu, *et al*., “Green Data Center Placement in Optical Cloud Networks,” *IEEE Transactions on Green Communications and Networking*, vol. 1, no. 4, pp. 347-357, 2017.