Distributed Algorithms for Content Allocation in Interconnected Content Distribution Networks

Valentino Pacifici and György Dán
ACCESS Linnaeus Center, School of Electrical Engineering
KTH, Royal Institute of Technology, Stockholm, Sweden
E-mail: {pacifici,gyuri}@kth.se

***Abstract*—Internet service providers increasingly deploy internal CDNs with the objective of reducing the traffic on their transit links and to improve their customers' quality of experience. Once ISP managed CDNs (nCDNs) become commonplace, ISPs would likely provide common interfaces to interconnect their nCDNs for mutual benefit, as they do with peering today. In this paper we consider the problem of using distributed algorithms for computing a content allocation for nCDNs. We show that if every ISP aims to minimize its cost and bilateral payments are not allowed then it may be impossible to compute a content allocation. For the case of bilateral payments we propose two distributed algorithms, the aggregate value compensation (AC) and the object value compensation (OC) algorithms, which differ in terms of the level of parallelism they allow and in terms of the amount of information exchanged between nCDNs. We prove that the algorithms converge, and we propose a scheme to ensure ex-post individual rationality. Simulations performed on a real AS-level network topology and synthetic topologies show that the algorithms have geometric rate of convergence, and scale well with the graphs' density and the nCDN capacity.**

I. Introduction

Commercial content distribution networks (CDNs) have for a decade dominated the market of digital media delivery. For content providers they offer relatively low delivery costs compared to investing in an own infrastructure, they provide dynamically scaling bandwidth to satisfy sudden surges of demand, and through multiple surrogate servers they provide better quality of experience (QoE) for customers than a system based on a single content delivery server [1], [2].

As the digital media delivery market matures, major over-the-top content providers, such as Netflix, Hulu, etc, try to maintain customer satisfaction through increasing QoE; 3D content has become commonplace, and super HD content has become available recently [3]. Increased QoE results in increased bitrates, which stresses network operators' networks, yet in the traditional CDN-based content distribution model network operators are not part of the revenue chain. At the same time, good QoE may also require control of the network resources between the CDN surrogate and the customers' premises and needs content to be placed closer to the customers.

Many network providers have started to deploy their own CDNs for the above reasons, and recent industry efforts aim to interconnect these network provider managed CDNs (nCDNs), potentially also with traditional commercial CDNs [2], [4]. For content providers, nCDN interconnection provides a transparent solution for bringing content closer to the customers than any single CDN would be able to provide. For network providers, nCDN interconnection can improve CDN availability and customer QoE.

As nCDNs often prefetch content based on predicted demands during periods of low demands (e.g., NetFlix Open Connect), successful nCDN interconnection requires that given predicted demands, the nCDNs be able to agree on a content allocation that serves all service providers' interests. In lack of a central authority the agreement has to be based on a distributed algorithm, the algorithm should not reveal confidential information, and the resulting allocation should be such that no nCDN fares worse due to interconnection, as otherwise nCDNs would have no incentive to interconnect.

In this paper we address the design of distributed algorithms for content allocation among interconnected CDNs. We propose a model of CDN interconnection assuming that CDNs aim to maximize the QoE of their customers, and we show that self-enforcing content allocations may not exist if payments are not allowed among nCDNs. We propose two distributed algorithms that use bilateral compensations to guarantee convergence to a content allocation and we propose an opt-out scheme, which combined with the two algorithms ensures that the resulting allocations are individually rational. Thus, participation according to the proposed algorithms is *ex-post individually rational* for all nCDNs. We use simulations on a measured Internet AS-level topology to evaluate the proposed algorithms, and we show that faster convergence can be achieved if nCDNs reveal more private information, such as content demands. To the best of our knowledge ours is the first work to consider the design of ex-post individually rational distributed algorithms for CDN interconnection.

The rest of the paper is organized as follows. Section II describes the system model. In Section III we show that a satisfactory content allocation may not exist without payments. In Section IV we design two distributed algorithms and we prove their convergence. Section V evaluates the proposed algorithms in terms of convergence rate and achieved cost savings. In Section VI we review the related work. Section VII concludes the paper.

II. System Model

We consider a set of autonomous service providers N. Each service provider manages a CDN; we refer to the CDN managed by service provider $i \in N$ as network CDN (nCDN) i. The customers of service provider i generate requests for content items from the set $\mathcal{O}$ of all content items. We make the common assumption that content is divisible in same-sized chunks, thus every item $o \in \mathcal{O}$ has the same size [5], [6]. The customers of service provider i generate requests for content item o at an average rate of $w_i^o \in \mathbb{R}_+$. We denote the set of content items stored by nCDN i by the set $A_i \in \mathcal{A}_i = \{A \subset \mathcal{O} : |A| = K_i\}$, where $K_i \in \mathbb{N}_+$ is the maximum number of items that nCDN i can store. In what follows, we use the terms *nCDN* and *service provider* interchangeably.

We model the relationships between the nCDNs by an undirected graph $\mathcal{G}(N, E)$, called the *interconnection graph*. There is an edge between nCDN i and nCDN j iff they are connected, and we use $\mathcal{N}(i) = \{j|(i,j) \in E\}$. We denote by A_{-i} the content item allocations of every nCDN other than nCDN i and by $\mathcal{R}_i(A_{-i})$ the set of items that can be retrieved from the nCDNs connected to nCDN i

$$\mathcal{R}_i(A_{-i}) \triangleq \bigcup_{j \in \mathcal{N}(i)} A_j. \tag{1}$$

We consider that each service provider aims to improve the quality of experience of its customers through decreasing the average access latency to content items. We denote by α_i the unit cost (i.e., access latency) of service provider i for serving an item stored in nCDN i, i.e., locally. If an item is not stored locally at nCDN i, it can be retrieved from one of the nCDNs $\mathcal{N}(i) \subset N$ connected to nCDN i. We denote by β_i^j the unit cost for serving an item from a connected nCDN $j \in \mathcal{N}(i)$. Observe that β_i^j depends on the neighbor j. If item o is available neither locally nor at a connected nCDN, it needs to be retrieved from the origin content provider in the network. We denote by γ_i the unit cost of retrieving an item from the origin content provider. We make the reasonable assumption that it is faster to access an item stored in the local nCDN than to retrieve it from a connected nCDN, and it is faster to retrieve an item stored in a connected nCDN than retrieving it from the content provider, i.e., $\alpha_i < \beta_i^j < \gamma_i$. This assumption is not restrictive, as if $\beta_i^j \geq \gamma_i$, we can remove (i,j) from E.

A. Average Access Latency Cost

We express the cost in terms of average access latency incurred by service provider i in allocation A as $C_i(A) = \sum_{o \in \mathcal{O}} C_i^o(A_i, A_{-i})$, where $C_i^o(A_i, A_{-i})$ is the cost for accessing item $o \in \mathcal{O}$,

$$C_i^o(A_i, A_{-i}) = w_i^o \begin{cases} \alpha_i & \text{if } o \in A_i \\ \min\limits_{j \in \mathcal{N}(i)} \{\beta_i^j | o \in A_j\} & \text{if } o \in \mathcal{R}_i(A_{-i}) \backslash A_i \\ \gamma_i & \text{otherwise.} \end{cases} \tag{2}$$

Observe that (i) the content allocations of the nCDNs in $\mathcal{N}(i)$ influence the cost of nCDN i through the set A_{-i}, and (ii) if item o is stored at several connected nCDNs then nCDN i retrieves it from the one with lowest unit cost. The cost incurred by service provider i for serving item o can be rewritten as

$$\begin{aligned} C_i^o(A_i, A_{-i}) &= C_i^o(\varnothing, A_{-i}) - (C_i^o(\varnothing, A_{-i}) - C_i^o(A_i, A_{-i})) \\ &= C_i^o(\varnothing, A_{-i}) - CS_i^o(A_i, A_{-i}), \end{aligned}$$

where $CS_i^o(A_i, A_{-i})$ is the cost saving that service provider i achieves by allocating item o given the content allocation at the nCDNs connected to nCDN i. Since the cost $C_i^o(\varnothing, A_{-i})$ is independent of the allocation A_i of nCDN i, finding the minimum cost is equivalent to finding the maximum aggregated cost saving

$$\begin{aligned} \arg\min_{A_i} C_i(A_i, A_{-i}) &= \arg\min_{A_i} \sum_o C_i^o(A_i, A_{-i}) \\ &= \arg\max_{A_i} \sum_o CS_i^o(A_i, A_{-i}). \end{aligned}$$

If nCDN i allocates item o, i.e., $o \in A_i$, then the cost saving can be rewritten as

$$CS_i^o(\{o\}, A_{-i}) = \begin{cases} w_i^o\,[\gamma_i - \alpha_i] & \text{if } o \notin \mathcal{R}_i(A_{-i}) \\ w_i^o\,[\beta_i^o(A_{-i}) - \alpha_i] & \text{if } o \in \mathcal{R}_i(A_{-i}) \end{cases} \tag{3}$$

where $\beta_i^o(A_{-i})$ is the lowest unit cost at which nCDN i can retrieve item o from a connected nCDN

$$\beta_i^o(A_{-i}) \triangleq \min_{j \in \mathcal{N}(i)} \{\beta_i^j | o \in A_j\}. \tag{4}$$

If instead $o \notin A_i$, the cost saving $CS_i^o(\{o\}, A_{-i}) = 0$. It follows that finding the minimum cost for service provider $i \in N$ corresponds to solving a knapsack problem where the values of the items are their cost savings given the allocations A_{-i} of the other nCDNs, and where the total weight is K_i.

B. Problem Statement

We consider that nCDNs would compute a content allocation based on predicted demands periodically, e.g., during epochs of low demand such as early mornings, and would perform prefetching to implement the allocation. We assume that bilateral payments are feasible, if necessary, and a payment $p_{i,j}$ would appear as an additive term in the cost function of nCDNs i and j. Payments can be settled periodically similar to peering agreements.

Without cooperation, i.e., if the set of connected nCDNs $\mathcal{N}(i) = \varnothing$ for every service provider i, service provider i would optimize the content allocation in nCDN i in isolation, and would prefetch the K_i items with highest demands. We denote the resulting allocation, which is *optimal in isolation*, by A_i^I. The corresponding cost is $C_i^I(A_i^I) = \sum_{o \in A_i^I} w_i^o \alpha_i + \sum_{o \in \mathcal{O} \backslash A_i^I} w_i^o \gamma_i$.

Cooperation could allow service providers to decrease their average access latency cost compared to isolation. For an allocation A we define the cost saving gain as

$$r_i(A) = \frac{C_i^I(\varnothing) - C_i(A)}{C_i^I(\varnothing) - C_i^I(A_i^I)} \tag{5}$$

where $C_i^I(\varnothing) = \sum_{o \in \mathcal{O}} w_i^o \gamma_i$ is the cost incurred by service provider i with no nCDN. We call an allocation A *individually rational* if $r_i(A) \geq 1$. Observe that service provider i benefits from cooperating only if $r_i(A) > 1$.

Since there is no central authority, cooperation requires a *distributed algorithm* that (i) requires information exchange between connected service providers only, (ii) reveals little private information such as content demands, and (iii) in a *finite* number of steps leads to a content allocation A that is *ex-post individually rational* for all service providers i.

III. Failure of Local Greedy Optimization

Without payments, a content allocation among interconnected nCDNs would have to let every nCDN i allocate content items that minimize its cost C_i, given the allocations of its connected nCDNs $\mathcal{N}(i)$. Such an allocation is self-enforcing, as no nCDN could gain by deviating from it. Modeling the interaction of nCDNs as a strategic game $\Gamma =< N, (\mathcal{A}_i)_{i \in N}, (U_i)_{i \in N} >$, where the utility of player i is the sum of its cost savings $U_i(A_i, A_{-i}) = \sum_o CS_i^o(A_i, A_{-i})$, such a content allocation corresponds to a pure strategy Nash equilibrium A^* of Γ, i.e., a set of allocations $(A_i^*)_{i \in N}$ such that

$$A_i^* \in \arg\max_{A_i} U_i(A_i, A_{-i}^*). \tag{6}$$

It is easy to see that A^* is individually rational.

Given an initial allocation of content items, $(A_i)_{i \in N}$, a distributed algorithm that might be used to compute A^* and one that reveals little private information is the *Local-Greedy* algorithm shown in Fig. 1. According to the *Local-Greedy* algorithm, at time step t a single nCDN i_t can update its allocation from $A_{i_t}(t-1)$ to an allocation $A_{i_t}(t)$ that increases its cost saving given the allocations of the other nCDNs $A_{-i_t}(t-1)$. The *Local-Greedy* algorithm requires little signaling: upon time step t nCDN i_t has to send the set $E_{i_t}(t) \triangleq A_{i_t}(t-1) \setminus A_{i_t}(t)$ of evicted and the set $I_{i_t}(t) \triangleq A_{i_t}(t) \setminus A_{i_t}(t-1)$ of inserted items to its neighboring nCDNs. The *Local-Greedy* algorithm terminates when no nCDN i can increase its cost saving by updating its allocation. By definition (6), if the *Local-Greedy* algorithm terminates, then the content allocation reached by the nCDNs is a pure strategy Nash equilibrium A^* of Γ. Nonetheless, it is not clear whether (i) an equilibrium allocation always exists and whether (ii) the *Local-Greedy* algorithm would lead to an equilibrium even if it exists.

At time step t:
- nCDN $i_t \in N$ chooses a content allocation $A_{i_t}(t)$ such that $CS_{i_t}(A_{i_t}(t), A_{-i_t}(t-1)) > CS_{i_t}(A(t-1))$.
- nCDN i_t communicates to $\forall j \in \mathcal{N}(i_t)$ the set of items it plans to evict $E_{i_t}(t)$ and insert $I_{i_t}(t)$.

Fig. 1: Pseudocode of the *Local-Greedy* algorithm.

In what follows we show that there are instances of the content allocation problem for which an equilibrium allocation A^* that satisfies (6) does *not exist*.

Non-Existence of Equilibrium Content Allocations

The strategic game Γ can be interpreted as a resource allocation game where the resources are the items, $c_i^o \triangleq w_i^o[\gamma_i - \alpha_i] \in \mathbb{R}_+$ is the value of resource o for player i and $0 < \delta_i^j \triangleq \frac{\beta_i^j - \alpha_i}{\gamma_i - \alpha_i} < 1$ is the penalty due to sharing the resource with player j. The expression of the cost saving in (3) becomes

$$CS_i^o(\{o\}, A_{-i}) = \begin{cases} c_i^o & \text{if } o \notin \mathcal{R}_i(A_{-i}) \\ c_i^o \min\limits_{j \in \mathcal{N}(i)} \{\delta_i^j | o \in A_j\} & \text{if } o \in \mathcal{R}_i(A_{-i}). \end{cases} \tag{7}$$

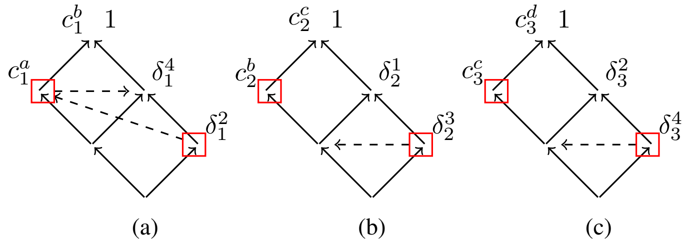

Fig. 2: Cost saving graphs of nCDNs 1 to 3 in Example 1. The squares show the cost savings of each nCDN given the content allocation of its neighbors in the content allocation (a, b, c, d, d).

Observe that the cost incurred by player i for retrieving item o depends on which neighboring players store item o, not only on whether any neighboring player stores it as in [7], [8], [9]. As a consequence, results on the existence of Nash Equilibria in player-specific graphical congestion games do not apply.

Consider the following example.

Example 1. *Consider nCDNs $N = \{1, \ldots, 5\}$ and the set $\mathcal{O} = \{a, b, c, d\}$ of content items. The nCDNs are interconnected according to the graph in Figure 3a. For nCDN 5*

$$c_5^d \delta_4^5 > c_5^o \quad \forall o \in \mathcal{O} \setminus \{d\}. \tag{8}$$

For nCDN 1 the demands and the costs satisfy

$$\delta_1^2 < \delta_1^4, \; c_1^a < c_1^b, \tag{9}$$

$$c_1^b \delta_1^2 < c_1^a < c_1^b \delta_1^4. \tag{10}$$

Inequalities (9-10) specify a lattice (a poset with least and greatest element) over the cost savings $CS_1^o(\{o\}, A_{-1})$, which is shown in Figure 2a; we call it the cost saving graph. *An arrow between two cost savings points towards the greater of the two.*

The lattice is on the one hand the product of two totally ordered sets (solid arrows): values $\{c_i^o \, | o \in \mathcal{O}\}$ and link costs $\{\delta_i^j | j \in \mathcal{N}(i)\} \cup \{1\}$. The greatest element of the lattice is the cost saving c_i^o of the item $o \in \mathcal{O}$ with highest rate w_i^o at nCDN i when it is not allocated by any connected nCDN $j \in \mathcal{N}(i)$, i.e. $o \notin \mathcal{R}_i(A_{-i}) \Rightarrow CS_i^o(\{o\}, A_{-i}) = c_i^o$. The least element of the lattice is the cost saving $c_i^p \delta_i^j$ of the item $p \in \mathcal{O}$ with lowest rate w_i^p when it is allocated by the connected nCDN $j \in \mathcal{N}(i)$ such that $j = \arg\min_{k \in \mathcal{N}(i)} \delta_i^k$. On the other hand, the lattice is specified through additional inequalities, such as (10) for nCDN 1 (dashed lines).

The cost saving graphs for nCDNs $i \in \{2, 3, 4\}$ are shown in Figure 2b, 2c and 3b, respectively. The squares in Figures 2 and 3b represent the cost savings $CS_i^o(\{o\}, A_{-i})$ of the corresponding nCDN $i \in \{1, 2, 3, 4\}$ at content allocation $A = (a, b, c, d, d)$. We omit the relations between cost savings that are not relevant for the example.

We are now ready to prove the following.

Theorem 1. *An equilibrium content allocation that simultaneously minimizes the cost of every nCDN (pure Nash equilibrium) may not exist.*

Proof. We prove the theorem by showing that the game described in Example 1 does not possess a Nash equilibrium.

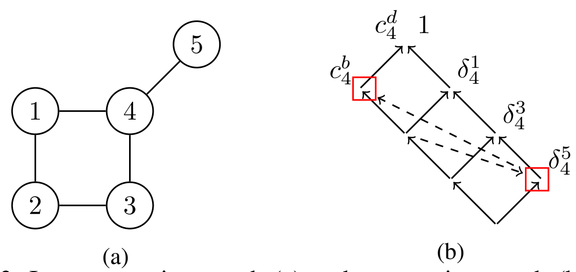

Fig. 3: Interconnection graph (a) and cost saving graph (b) of nCDN 4 in Example 1. The squares show the cost savings of nCDN 4 in the content allocation (a, b, c, d, d).

From (8) it follows that no content allocation A such that $A_5 \neq \{d\}$ is a Nash equilibrium. We can furthermore restrict the action set of each nCDN to the items that are included in its cost saving graph, according to the following

$$\mathcal{A}_1 = \{\{a\}, \{b\}\} \quad \mathcal{A}_2 = \{\{b\}, \{c\}\}$$
$$\mathcal{A}_3 = \{\{c\}, \{d\}\} \quad \mathcal{A}_4 = \{\{d\}, \{b\}\}$$

This results in a total of 16 possible content allocations. From the cost saving graphs in Figures 2 and 3b it follows that any content allocation where two interconnected nCDNs store the same item is not a Nash equilibrium; there are 12 such allocations. Therefore, it is enough to focus on the 4 content allocations in which there is no pair of interconnected nCDNs that store the same item.

The following sequence of content allocations starts with these four content allocations (marked with bold) and shows a cycling sequence of updates of nCDNs that follow the *Local-Greedy* algorithm. We omit nCDN 5, which always stores item (d).

$$\begin{aligned}(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{d}) \underset{4}{\rightarrow} \ (\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{b}) \underset{3}{\rightarrow} \ (\boldsymbol{a},\boldsymbol{b},\boldsymbol{d},\boldsymbol{b}) \underset{2}{\rightarrow} \ (\boldsymbol{a},\boldsymbol{c},\boldsymbol{d},\boldsymbol{b}) \\ \underset{1}{\rightarrow} \ (b,c,d,b) \underset{4}{\rightarrow} \ (b,c,d,d) \underset{3}{\rightarrow} \ (b,c,c,d) \\ \underset{2}{\rightarrow} \ (b,b,c,d) \underset{1}{\rightarrow} \ (a,b,c,d)\end{aligned} \tag{11}$$

There is thus no NE, which proves the theorem. $\square$

Observe that so far we have not assumed any relationship between the link costs δ_i^j and δ_j^i of interconnected nCDNs. There are recent measurement studies [10] that show that it may be reasonable to assume that link costs are symmetric between interconnected nCDNs, i.e., $\delta_i^j = \delta_j^i \ \forall j \in \mathcal{N}(i)$. In the case of Example 1, the requirement of symmetric link costs implies a feasible total order on the link costs, $1 > \delta_1^4 > \delta_1^2 > \delta_2^3 > \delta_3^4 > \delta_4^5$, which leads to the following.

Corollary 1. *An equilibrium content allocation that simultaneously minimizes the cost of every nCDN may not exist even if link costs are symmetric.*

A corresponding non-existence result for the case of a linear cost function and a directed interconnection graph was provided in [11]. Observe that, in our model, a link (i, j) is effectively directed if δ_i^j is sufficiently smaller than δ_j^i, such that the content allocated at nCDN i does not affect the allocation that minimizes the cost function of nCDN j. The importance of Corollary 1 is that it extends the non-existence results to undirected interconnection graphs.

Since an equilibrium allocation may not exist, Theorem 1 and Corollary 1 imply that if payments are not allowed then cost minimizing nCDNs may not be able to compute a content allocation, and algorithms like *Local-Greedy* may never terminate. Since it is infeasible to determine a-priori whether an equilibrium allocation exists (for reasons of computational complexity and because doing so would require global knowledge), computing a content allocation must involve payments to guarantee finite execution time.

IV. Bilateral Compensation-based Allocation

A natural solution involving payments would be to model the problem as a coalition game with transferable utility, define the value function of a coalition as the maximum cost saving achievable by the set of players that form the coalition, and use the Shapley value for computing compensations. Since this value function is super-additive, the Shapley value is individually rational [12]. Nonetheless, such a solution is infeasible for several reasons. Computing the value of a coalition requires a single entity to know all item popularities. Second, it follows from the cost function (2) that computing the value of a coalition is NP-hard, and the computation needs to be done for all coalitions that induce a connected subgraph in $\mathcal{G}$.

In the following, we propose two distributed algorithms that involve *bilateral* compensations for computing an individually rational content allocation. The two algorithms differ in the amount of revealed private information, in the level of parallelism that they allow and, as we will see, in terms of convergence rate.

A. Aggregate-value Compensation Algorithm

Following the aggregate value compensation (AC) algorithm, at every time step t there is a set N_t of nCDNs that is allowed to update its content allocation. Given an allocation of content items $A(t-1)$, an update made by nCDN $i_t \in N_t$ from $A_{i_t}(t-1)$ to $A_{i_t}(t)$ can result in an increase of the cost (2) for one or more connected CDNs $j \in \mathcal{N}(i_t)$. According to the AC algorithm, an nCDN $j \in \mathcal{N}(i_t)$, $i_t \in N_t$, that would suffer an increase of cost $C_j(A_{N_t}(t), A_{-N_t}(t-1)) > C_j(A(t-1))$, offers a compensation to an nCDN $i_t \in \mathcal{N}(j) \cap N_t$ equal to its cost increase $\Delta C_j(t) \triangleq C_j(A_{i_t}(t), A_{-i_t}(t-1)) - C_j(A(t-1))$. We use $D_t \subseteq \mathcal{N}(i_t)$ to denote the set of nCDNs that offer a compensation,

$$j \in D_t \Leftrightarrow \Delta C_j(t) > 0. \tag{12}$$

The compensations are used to deter nCDNs from performing updates: nCDN $i_t \in N_t$ performs the update despite the offered compensation if the aggregate compensation offered by all connected nCDNs is lower than the gain it achieves from updating the content allocation from $A_{i_t}(t-1)$ to $A_{i_t}(t)$. We call this the *Aggregate-value Compensation* (AC) algorithm, and we summarize it in Figure 4. Observe that the AC algorithm does not specify how N_t is chosen at each time step t, nor how an nCDN $j \in D_t$ chooses the recipient nCDN i_t of its compensation.

Before proving convergence for specific choices of N_t, we make the following definition.

At time step t:

- Every nCDN $i_t \in N_t$ computes a content allocation $A^P_{i_t}(t)$ s.t. $CS_{i_t}(A^P_{i_t}(t), A_{-i_t}(t-1)) > CS_{i_t}(A(t-1))$.
- Every nCDN $i_t \in N_t$ communicates to $\forall j \in \mathcal{N}(i_t)$ the set of items it plans to evict $E_{i_t}(t)$ and insert $I_{i_t}(t)$.
- Every nCDN $j \in \mathcal{N}(N_t)$ s.t. $\Delta C_j(t) > 0$ offers a compensation $p^{i_t}_j(t) = \Delta C_j(t)$ to one nCDN $i_t \in \mathcal{N}(j) \cap N_t$
- **If** $\sum_{j\in D_t} p^{i_t}_j(t) \geq -\Delta C_{i_t}(t)$, then nCDN i_t accepts the compensation and it does not make the update, i.e., $A_{i_t}(t) = A_{i_t}(t-1)$.
  Otherwise nCDN i_t refuses the compensation and updates its allocation from $A_{i_t}(t-1)$ to $A_{i_t}(t) = A^P_{i_t}(t)$.
- Every nCDN $i_t \in N_t$ communicates to $\forall j \in \mathcal{N}(i_t)$ its decision, i.e. whether $A_{i_t}(t) = A_{i_t}(t{-}1)$ or $A_{i_t}(t) = A^P_{i_t}(t)$.

Fig. 4: Aggregate-value Compensation (AC) Algorithm

Definition 1. A sequence $N_t \subseteq N, t = 1, \ldots$ of sets of nCDNs is a complete sequence, if for all t and each nCDN $i \in N$ there exists a time step $t' > t$ such that $i \in N_{t'}$.

Asynchronous operation: Let us first consider that only one nCDN $i_t \in N_t$ is allowed to update its allocation at each time slot t. Thus, the sets N_{t_1}, N_{t_2}.. are singletons and nCDN i_t is the only recipient of the compensation of each nCDN $j \in D_t$. The following result shows that the AC algorithm converges if used asynchronously.

Theorem 2. *Let N_t be a complete sequence of singleton sets and every nCDN use the* AC *algorithm. We refer to this as the* 1-AC *algorithm. The* 1-AC *algorithm converges to an allocation $\boldsymbol{A}$ of content items to interconnected nCDNs in a finite number of time steps.*

Proof. We prove the theorem by showing that the aggregate cost $C(A) \triangleq \sum_{i\in N} C_i(A_i, A_{-i})$ strictly decreases at every update made by any nCDN following the 1-AC algorithm.

Consider a nCDN i_t that updates its content allocation from $A_{i_t}(t-1)$ to $A_{i_t}(t)$ at time step t. It follows from (2) that $C_k(A(t)) = C_k(A(t-1))$ for any nCDN $k \notin \mathcal{N}(i_t)$. We can calculate the aggregate cost function $C(A(t))$ after the update of nCDN i_t as

$$C(A(t)) = C(A(t{-}1)) + \Delta C_{i_t}(t) + \sum_{j\in D_t} \Delta C_j(t) + \sum_{j\in\mathcal{N}(i_t)\setminus D_t} \Delta C_j(t).$$

From (12) it follows that $\sum_{j\in\mathcal{N}(i_t)\setminus D_t} \Delta C_j(t) \leq 0$. Moreover, since nCDN i_t refused the compensation offered by the connected nCDNs in D_t, it follows that $\Delta C_{i_t}(t) + \sum_{j\in D_t} \Delta C_j(t) < 0$. Hence, at every update of the 1-AC algorithm $C(A(t)) < C(A(t-1))$. Since the set of all content allocations is finite and the sequence N_t is complete, this proves the theorem. $\square$

A significant shortcoming of the 1-AC algorithm is that it requires global synchronization. Furthermore, if nCDN i_t is chosen uniformly at random at every time step t, the probability that nCDN i_t can decrease its cost C_{i_t} by updating its content allocation $A_{i_t}(t-1)$ at time step t decreases as the 1-AC approaches allocation $\boldsymbol{A}$. As a consequence, the convergence of the 1-AC algorithm may be slow.

Plesiochronous operation: In the following we show that convergence can be guaranteed even if the sets N_t are not singletons. Before we formulate our result, we recall the following definition from graph theory.

Definition 2. A *k-independent set* $\mathcal{I}^k$ of a graph $\mathcal{G} = (N, E)$ is a subset $\mathcal{I}^k \subseteq N$ of the vertexes of $\mathcal{G}$ such that the distance between any two vertexes of $\mathcal{I}^k$ in $\mathcal{G}$ is at least $k{+}1$. We denote by $\mathfrak{I}^k$ the set of all the k-independent sets of the interconnection graph $\mathcal{G}$.

We can now prove the following.

Theorem 3. *Let N_t be a complete sequence of 2-independent sets and every nCDN use the AC algorithm. We refer to this as the $\mathcal{I}^2$-*AC *algorithm. The $\mathcal{I}^2$-*AC *algorithm converges to an allocation $\boldsymbol{A}$ of content items to interconnected nCDNs in a finite number of time steps.*

Proof. Consider a nCDN $j \in \mathcal{N}(i_t)$, connected to $i_t \in \mathcal{I}^2_t$. From the definition of 2-independent set follows that i_t is the only nCDN in $\mathcal{N}(j)$ that is allowed to update its content allocation A_{i_t} at time step t. Hence, it is possible to compute the aggregate cost function $C(A(t))$ from $C(A(t-1))$ as follows

$$C(A(t)) = C(A(t-1)) \;+ \\ + \sum_{i_t\in U_t} \Big(\Delta C_{i_t}(t) + \sum_{j\in D_t} \Delta C_j(t) + \sum_{j\in\mathcal{N}(i_t)\setminus D_t} \Delta C_j(t)\Big),$$

where $U_t \subseteq \mathcal{I}^2_t$ is the set of nCDNs i_t such that $A_{i_t}(t) \neq A_{i_t}(t{-}1)$. From the same argument in the proof of Theorem 2 it follows that $C(A(t)) < C(A(t-1))$ at every update of the $\mathcal{I}^2$-AC algorithm. $\square$

Since the 2-independent sets of $\mathcal{G}$ are typically small, the number of nodes that can make updates simultaneously is small, and thus the convergence rate of the $\mathcal{I}^2$-AC algorithm may be only marginally faster than that of 1-AC. The number of simultaneous updates could be increased by using 1-independent sets, i.e., $\mathcal{I}^1$-AC, but the convergence of the $\mathcal{I}^1$-AC algorithm can not be guaranteed. We therefore propose an alternative to the AC algorithm in the following.

B. Object-value Compensation Algorithm

The object value compensation (OC) algorithm, shown in Figure 5, is similar to the AC algorithm, the difference is that nCDNs offer a compensation for each individual object that is to be evicted, instead of offering a compensation for the set of objects to be evicted. As we will see this difference allows for significantly faster convergence, but at the price of revealing more information about content item popularities.

For the OC algorithm we can prove the following.

Theorem 4. *Let N_t be a complete sequence of 1-independent sets and every nCDN use the* OC *algorithm. We refer to this as the $\mathcal{I}^1$-*OC *algorithm. The $\mathcal{I}^1$-*OC *algorithm converges to an allocation $\boldsymbol{A}$ of content items to interconnected nCDNs in a finite number of time steps.*

At time step t:

- Every nCDN $i_t \in N_t$ computes a content allocation $A^P_{i_t}(t)$ such that $\boldsymbol{CS}_{i_t}(A^P_{i_t}(t), A_{-i_t}(t-1)) > \boldsymbol{CS}_{i_t}(A(t-1))$.
- Every nCDN $i_t \in N_t$ communicates to $\forall j \in \mathcal{N}(i_t)$ the set of items it plans to evict $E_{i_t}(t)$ and insert $I_{i_t}(t)$.
- Every nCDN $j \in \cup_{i_t \in N_t} \mathcal{N}(i_t)$ calculates
$$\Delta\widetilde{C}^o_j(t) = C^o_j(A^P_{N_t}(t), A_{-N_t}(t-1)) - C^o_j(A(t-1))$$
for all $o \in \cup_{N_t} E_{i_t}$. If $\Delta\widetilde{C}^o_j(t) > 0$, nCDN j offers to a nCDN $k \in N_t$ such that $o \in E_k$ and $\beta^k_j = \beta^o_j(A_{-k}(t))$ a compensation $p^o_{j,k}(t) \triangleq \Delta\widetilde{C}^o_j(t)$.
- **If** $\sum_{j \in D_{i_t}} \sum_{o \in E_{i_t}} p^o_{j,i_t}(t) \geq -\Delta C_{i_t}(t)$, then nCDN i_t accepts the offer and it does not make the update, i.e., $A_{i_t}(t) = A_{i_t}(t-1)$.
Otherwise nCDN i_t refuses the offer and updates its allocation from $A_{i_t}(t-1)$ to $A_{i_t}(t) = A^P_{i_t}(t)$.
- Every nCDN $i_t \in N_t$ communicates to $\forall j \in \mathcal{N}(i_t)$ its decision, i.e. whether $A_{i_t}(t) = A_{i_t}(t–1)$ or $A_{i_t}(t) = A^P_{i_t}(t)$.

Fig. 5: Object-value Compensation (OC) Algorithm

Proof. Consider the compensation $p^o_{j,k}(t)$ offered by nCDN j to nCDN k for the eviction of item $o \in E_k$ at time step t. Substituting (3) in the expression of $\Delta\widetilde{C}^o_j(t)$ we obtain

$$p^o_{j,k}(t) = w^o_j \left[\beta^o_j\left((A^P_{\mathcal{I}^1_t}(t), A_{-\mathcal{I}^1_t}(t-1))\right) - \beta^o_j\left(A(t-1)\right)\right].$$

We call U_t the set of nCDNs that update their content allocation at time step t of the algorithm, i.e. $U_t = \{i_t \in \mathcal{I}^1_t | A_{i_t}(t) \neq A_{i_t}(t-1)\}$. Since $U_t \subseteq \mathcal{I}^1_t$, it follows from (4) that $\beta^o_j\left((A^P_{\mathcal{I}^1_t}(t), A_{-\mathcal{I}^1_t}(t-1))\right) \geq \beta^o_j\left((A^P_{U_t}(t), A_{-U_t}(t-1))\right)$ and thus

$$p^o_{j,k}(t) \geq C^o_j(A(t)) - C^o_j(A(t-1)). \quad (13)$$

In the following we use (13) to prove that $C(A(t)) < C(A(t-1))$ at every update of the $\mathcal{I}^1$-OC algorithm. We can express the aggregate cost change $\Delta C(t) = C(A(t)) - C(A(t-1))$ as

$$\Delta C(t) = \sum_{i_t \in U_t} \Delta C_{i_t}(t) + \sum_{j \in D_t} \Delta C_j(t) + \sum_{j \in \mathcal{N}(i_t) \setminus D_t} \Delta C_j(t). \quad (14)$$

From (13) it follows that the second term

$$\sum_{j \in D_t} \Delta C_j(t) = \sum_{j \in D_t} \sum_{o \in \mathcal{O}} \Delta C^o_j(t) \leq \sum_{j \in D_t} \sum_{o \in \mathcal{O}} p^o_{j,k}(t). \quad (15)$$

Substituting (15) into (14) we obtain

$$\begin{aligned}\Delta C(t) &\leq \sum_{i_t \in U_t} \Delta C_{i_t}(t) + \sum_{j \in D_t} \sum_{o \in \mathcal{O}} p^o_{j,k}(t) \\ &= \sum_{i_t \in U_t} \Big(\Delta C_{i_t}(t) + \sum_{j \in D_{i_t}} \sum_{o \in \mathcal{O}} p^o_{j,i_t}(t)\Big).\end{aligned}$$

Since every nCDN $i_t \in U_t$ refused the offer and updated its allocation, it holds that $\Delta C_{i_t}(t) + \sum_{j \in D_{i_t}} \sum_{o \in \mathcal{O}} p^o_{j,i_t}(t) < 0$ for all $i_t \in U_t$. Since the set of all content allocations is finite and the sequence N_t is complete, this proves the theorem. □

C. Achieving Individual Rationality

The proposed algorithms terminate in a finite number of time steps, but the resulting content allocation $\boldsymbol{A}$ may not be individually rational, i.e., for some nCDNs i it may hold $r_i(\boldsymbol{A}) < 1$. The nCDNs $i \in \{i \in N | r_i(\boldsymbol{A}) < 1\}$ would not have an incentive to implement $\boldsymbol{A}_i$, and would instead implement A^I_i. The OPT OUT scheme, shown in Figure 6, allows these nCDNs to implement A^I_i instead of $\boldsymbol{A}_i$ and iteratively re-executes the distributed algorithm with the remaining nCDNs; hence the final allocation is individually rational.

1) Set $\ell \leftarrow 1$ and $N^\ell_c \leftarrow N$
2) At round ℓ:
   - The nCDNs in N^ℓ_c run algorithm AC or OC until it terminates, in allocation $\boldsymbol{A}$.
   - Set $A^\ell \leftarrow \boldsymbol{A}$ and $N^{\ell+1}_c \leftarrow \{i \in N^\ell_c | r_i(A^\ell) \geq 1\}$.
3) If $|N^\ell_c \setminus N^{\ell+1}_c| > 0$:
   - Set $\ell \leftarrow \ell + 1$ and go to step 2).

Fig. 6: OPT OUT scheme

Corollary 2. *The* OPT OUT *scheme reaches an individually rational content allocation $\bar{A}$ in a finite number of iterations.*

Proof. Observe that the OPT OUT scheme terminates in allocation $\bar{A} = (A^\ell_{N^\ell_c}, A^I_{N \setminus N^\ell_c})$ only if $N^\ell_c = N^{\ell+1}_c$ at step 3). This is true if either $|N^\ell_c| = 0$ or $r_i(A^\ell) < 1 \ \forall i \in N^\ell_c$. In both cases $\bar{A}$ is individually rational. □

Thus the $\mathcal{I}^2$-AC and the $\mathcal{I}^1$-OC algorithms combined with the OPT-OUT scheme are ex-post individually rational distributed algorithms for computing content allocations in a finite number of time steps, without prior global knowledge of the item popularities.

V. EVALUATION

We use simulations to validate the results in Section IV and to evaluate the convergence rate and the achieved gains for the cooperating nCDNs.

We consider three network topologies for the evaluation. The first topology is based on the Internet's AS-level topology reported in the CAIDA dataset [13] as of 1 Nov. 2013. In order to have a fairly large interconnection graph, we consider the ASes in the CAIDA dataset that are in Europe. As very small ASs are unlikely to deploy their own CDNs, we only consider ASs that have more than 2^{16} IP addresses allocated. We consider two ASes connected if they have a business relationship (peering or transit) reported in the CAIDA dataset. We call *CAIDA graph* the largest connected component of the resulting topology, which consists of 638 ASes with an average node degree of 10.77. The other two topologies are Erdős-Rényi (CAIDA-ER) and Barabási-Albert (CAIDA-BA) random graphs that have same number of vertexes, average node degree and node degree ranking as the CAIDA graph. The node degree distributions of the three topologies do, however, differ in terms of their skeweness. We computed distance-1 and distance-2 colorings of all graph topologies by using the Welsh-Powell [14] and the Lloyd-Ramanathan [15] algorithms, respectively. We used $\alpha_i = 0.5$, $\gamma_i = 20$ at every nCDN and we computed the β^j_i as the propagation delay between nCDNs i

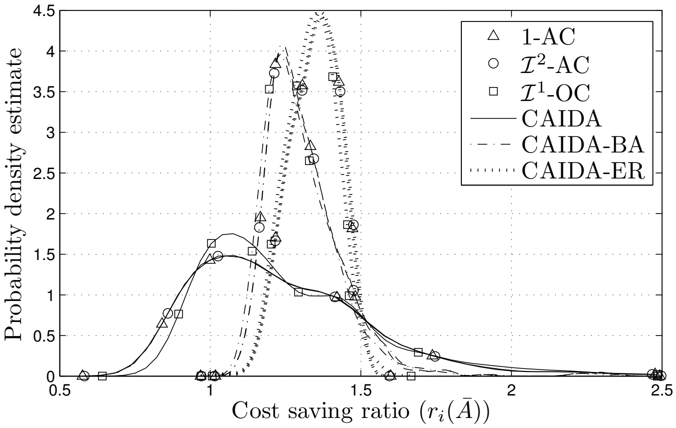

Fig. 7: Probability density estimate of $r_i(\bar{A})$ for the three algorithms 1-AC, $\mathcal{I}^2$-AC and $\mathcal{I}^1$-OC on the CAIDA, CAIDA-BA and CAIDA-ER graphs. Results from 400 simulations.

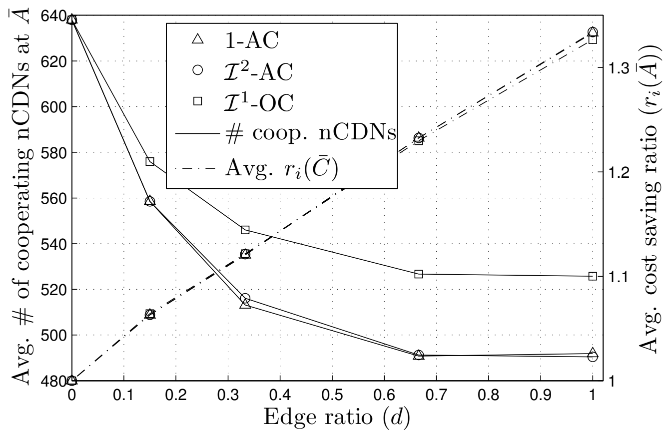

Fig. 8: Average number of nCDNs choosing to cooperate and average cost saving ratio $r_i(\bar{A})$ at allocation $\bar{A}$, as a function of the edge ratio d for the three algorithms on the CAIDA graph.

and j assuming a signal propagation speed of $2 \cdot 10^5$ km/s. We considered $|\mathcal{O}| = 3000$ objects and the demands w_i^o for the content items at the various nCDNs follow Zipf's law with exponent 1. To simulate the algorithms, at each time step we choose an nCDN or a k-independent set uniformly at random, thus the sequence is complete. If not otherwise specified, the results shown are the averages of 200 simulations and $K_i = 20$ for every nCDN $i \in N$. We omit the confidence intervals as the results are within 5% of the average at a 0.95 confidence level.

A. Individual Rationality

We start with considering the gain of cooperation and the necessity of the OPT OUT scheme. Figure 7 shows the probability density estimate of the cost saving ratios $r_i(A^{\ell=1})$ at the end of the first round of the OPT OUT scheme ($\ell = 1$) for all nCDNs for the three interconnection graphs and the three algorithms. The results show that the share of nCDNs for which the allocation is individually rational after the first round is determined by the graph topology. For the CAIDA-ER and the CAIDA-BA graphs, the content allocation is individually rational, $r_i(A^{\ell=1}) \geq 1$, for all nCDNs, and thus the OPT OUT scheme terminates after the first round, i.e. $\bar{A} = A^{\ell=1}$. On the contrary, for the CAIDA graph for many nCDNs $r_i(A^{\ell=1}) < 1$ after the first round. The difference is due to that the degree distribution of the CAIDA-BA graph is the most right-skewed among all the interconnection graphs, while the degree distribution of the CAIDA-ER graph is not skewed.

Observe that the probability densities for the 1-AC and $\mathcal{I}^2$-AC algorithms overlap, and are similar to that for the $\mathcal{I}^1$-OC algorithm. This suggests that the choice of the algorithm seems to have little impact on the gain from cooperation achieved by the nCDNs.

We evaluate the sensitivity of the results on synthetic topologies based on the CAIDA graph. The synthetic topologies were created by removing all edges from the CAIDA graph, and then reintroducing d fraction of the edges at random; the probability of reintroducing an edge between ASes i and j was proportional to the product of the number of IP addresses allocated to AS i and j.

Figure 8 shows the number of nCDNs choosing to cooperate and the average cost saving ratio $r_i(\bar{A})$ for the allocation $\bar{A}$ reached by the OPT OUT scheme, for the algorithms 1-AC, $\mathcal{I}^2$-AC and $\mathcal{I}^1$-OC on the CAIDA graph. The figure shows that the number of cooperating nCDNs is a decreasing convex function of the edge ratio d, suggesting that the majority of the nCDNs would not opt out from cooperation even if the graph was denser. Furthermore, the number of nCDN that would not opt out from cooperation is about 6% higher for the $\mathcal{I}^1$-OC algorithm compared to the $\mathcal{I}^2$-AC algorithm. At the same time the average cost saving ratio increases linearly, which is due to that the nCDNs have access to a linearly increasing amount of storage at neighbors.

B. Convergence Rate

We characterize the rate of convergence of the 1-AC, $\mathcal{I}^2$-AC and $\mathcal{I}^1$-OC algorithms by comparing the number of time steps needed to reach allocation A^ℓ during one round ℓ of the OPT OUT scheme. The number of time steps needed to reach A^ℓ is proportional to the time required by the algorithms to converge, as it also captures the parallelism embedded in the plesiochronous $\mathcal{I}^2$-AC and $\mathcal{I}^1$-OC algorithms.

Figure 9 shows the complementary CDF of the number of time steps needed to reach allocation A^ℓ based on 400 simulations for the three algorithms on the CAIDA, CAIDA-BA and CAIDA-ER graphs. The tail of each distribution decreases exponentially or faster as the number of time steps increases, which suggests that the rate of convergence is geometric. As expected, the 1-AC algorithm performs worst in terms of convergence rate, as it does not allow the nCDNs to update their allocations simultaneously. $\mathcal{I}^2$-AC and $\mathcal{I}^1$-OC are up to two orders of magnitude faster than 1-AC. Note that the fast convergence of the $\mathcal{I}^1$-OC algorithm is achieved at the price of increased information exchange between connected nCDNs compared to the 1-AC and $\mathcal{I}^2$-AC algorithms. In practice, the object-wise information exchange between ASs may be problematic due to privacy concerns.

Figure 10 shows that the average number of time steps needed to reach allocation A^ℓ is an increasing concave function of

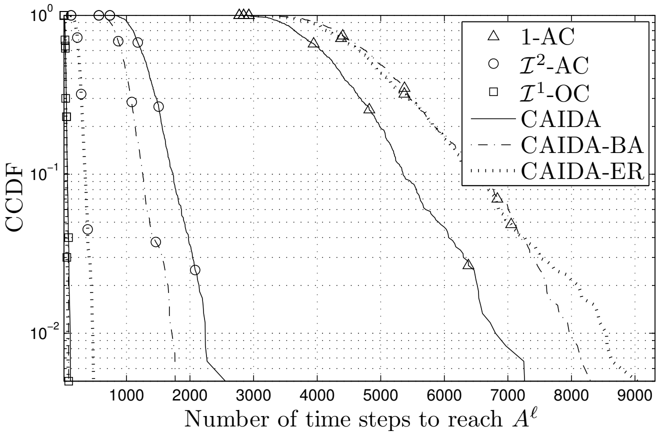

Fig. 9: Complementary CDF of the number of time steps needed to reach allocation A^ℓ for the algorithms 1-AC, $\mathcal{I}^2$-AC and $\mathcal{I}^1$-OC on the CAIDA, CAIDA-BA and CAIDA-ER graphs.

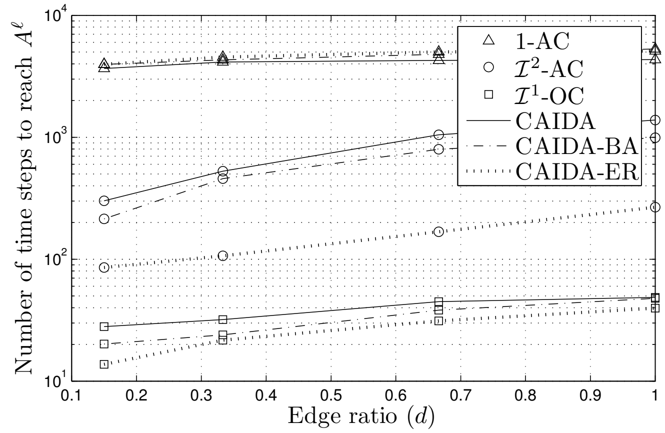

Fig. 10: Average number of time steps needed to reach allocation A^ℓ as a function of the edge ratio d for the three algorithms on the CAIDA, CAIDA-BA and CAIDA-ER graphs.

the edge ratio d for all algorithms and interconnection graphs. Observe that the three interconnection graphs rank analogously for algorithms $\mathcal{I}^2$-AC and $\mathcal{I}^1$-OC but not for algorithm 1-AC. The reason lies in the average sizes of the k-independent sets used by the algorithms, which are reported in Table I. The higher the average size of the k-independent sets, the higher the parallelism achieved by the $\mathcal{I}^k$-COMP algorithm, and the faster the convergence. As the coloring of the interconnection graph does not affect the performance of the 1-AC algorithm, the rankings of the three curves in both Figures 9 and 10 reflect other characteristics of the different network topologies.

C. Scaling for Storage Capacity

In the following we investigate the effect of increasing the storage capacity K_i on the convergence rate of the proposed algorithms. Figure 11 shows the average number of time steps to reach allocation A^ℓ during one round ℓ of the OPT OUT scheme. We plot one curve for each algorithm on each of the CAIDA and CAIDA-BA graphs, as a function of the storage capacity K_i. The convergence rate is surprisingly insensitive to the storage capacity and the algorithms rank analogously to Figure 10. To explain this insensitivity we plot the average number of content item updates performed by the nCDNs for the same algorithms and graphs in Figure 12. We make two observations. First, the number of content item updates is the same for 1-AC and $\mathcal{I}^2$-AC, as they are both based on aggregate value compensation. The nCDNs perform less updates using the $\mathcal{I}^1$-OC algorithm, as they exchange object-wise compensations. The nCDNs perform less updates using the $\mathcal{I}^1$-OC algorithm, as they exchange object-wise compensations. Second, an nCDN can do an arbitrary number of content item updates during one time step, thus although the number of content items increases for larger storage sizes, this does not result in slower convergence in Figure 11.

Graph	#1-ind. sets	avg. size	#2-ind. sets	avg. size
CAIDA	16	39.8	219	2.9
CAIDA-BA	10	63.8	131	4.9
CAIDA-ER	8	79.8	36	17.8

TABLE I: Number of k-independent sets and corresponding average size for the CAIDA, CAIDA-BA and CAIDA-ER interconnection graphs.

VI. Related Work

Our work is related to recent works on content placement in the context of CDNs [1], [16], [17], [18]. The majority of these works assume a single CDN operator and optimize content placement given a single performance objective. The authors in [16], [17] considered centralized algorithms for content placement and compared the retrieval cost for the different algorithms. Recently, [18] considered distributed algorithms that optimize for a single performance objective and provided analytical results for tree networks. In contrast to these works, in this paper we consider distributed algorithms for operator-managed CDNs, and thus the allocations need to be individually rational.

Closely related to ours are recent works on distributed selfish replication. Game theoretical analyses of equilibria and convergence for distributed selfish replication were considered in [7], [19], [20], [21], [22], [9]. The authors in [7] showed the existence of equilibria when the access costs are homogeneous and nodes form a complete graph. Similarly, [19], [20] assumed homogeneous costs and calculated the social cost of equilibria. The latter works considered that the nodes had no restriction on where to retrieve the content from. Other works [21], [22], [9] relax this assumption and introduce an interconnection graph to restrict the interaction between nodes. [21] assumed unit storage capacity and an infinite number of objects, showed the existence of equilibria and analyzed the price of anarchy for some special cases. [22] considered a variant of the problem where nodes can replicate a fraction of objects, and showed the existence of equilibria. The authors in [9] showed results in terms of convergence to equilibria in the case of homogeneous neighbor costs. In this paper we show that equilibrium existence results cannot be extended to the general problem of content replication on graphs and propose compensation-based algorithms that are guaranteed to converge.

Individually rational allocation of costs and revenues is the subject of cooperative game theory. Solution concepts such as the Shapley value and the core have found application in

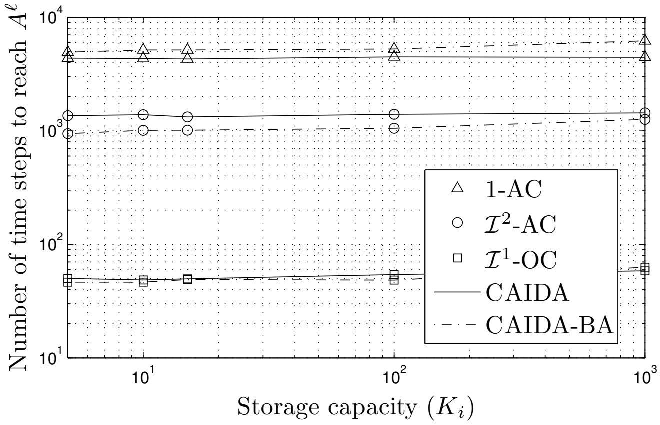

Fig. 11: Average number of time steps needed to reach allocation A^ℓ as a function of the storage capacity K_i for the algorithms 1-AC, $\mathcal{I}^2$-AC and $\mathcal{I}^1$-OC on the CAIDA and CAIDA-BA graphs.

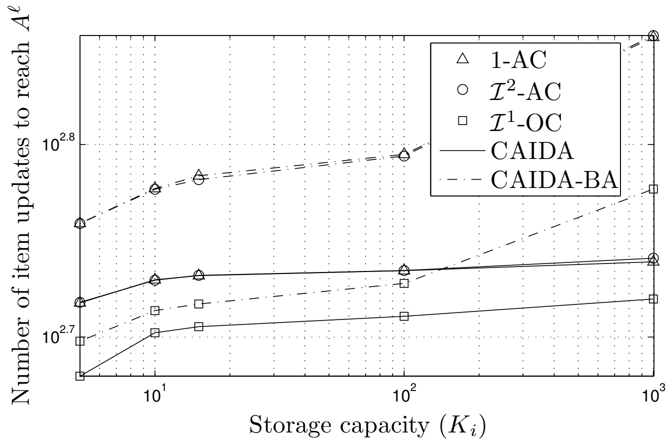

Fig. 12: Average number of content item updates needed to reach allocation A^ℓ as a function of the storage capacity K_i for the three algorithms on the CAIDA and CAIDA-BA graphs.

Internet routing [23] and in resource allocation [24], but these solution concepts require complete information and global enforcement, which make them impractical in the considered scenario. To the best of our knowledge this is the first work that proposes ex-post individually rational distributed algorithms for interconnected CDNs.

VII. Conclusion

We considered the problem of computing a content allocation among interconnected autonomous CDNs. We showed that without payments there may be no allocation that minimizes the cost of all CDNs, and thus payments are necessary to guarantee that greedy algorithms would converge. For the case that payments are possible, we proposed two bilateral compensation-based distributed algorithms that are ex-post individually rational. The two algorithms require different amounts of information to be revealed by the CDNs, and allow different levels of parallelism. Numerical results show that the algorithms have geometric convergence, and that if CDNs reveal more private information about their content demands, the convergence of the algorithms becomes faster. Our results also show that the convergence times are fairly insensitive to the graph density and the amount of CDN storage.

References

[1] A.-M. K. Pathan and R. Buyya, "A Taxonomy and Survey of Content Delivery Networks," The University of Melbourne, Tech. Rep., 2007.

[2] E. Bertrand, E. Stephan, T. Burbridge, P. Eardley, K. Ma, and G. Watson, "Use Cases for Content Delivery Network Interconnection," Internet Engineering Task Force (IETF), RFC, 2012. [Online]. Available: http://tools.ietf.org/html/rfc6770

[3] "Netflix." [Online]. Available: http://www.netflix.com

[4] L. Peterson and B. Davie, "Framework for CDN Interconnection," Internet Engineering Task Force (IETF), RFC, 2013. [Online]. Available: http://datatracker.ietf.org/doc/draft-ietf-cdni-framework/

[5] E. J. Rosensweig, J. Kurose, and D. Towsley, "Approximate Models for General Cache Networks," in *Proc. of IEEE INFOCOM*, 2010, pp. 1–9.

[6] C. Fricker, P. Robert, and J. Roberts, "A versatile and accurate approximation for LRU cache performance," in *Proc. of the 24th International Teletraffic Congress*, 2012.

[7] N. Laoutaris, O. Telelis, V. Zissimopoulos, and I. Stavrakakis, "Distributed selfish replication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 12, pp. 1401–1413, 2006.

[8] V. Pacifici and G. Dán, "Content-peering Dynamics of Autonomous Caches in a Content-centric Network," in *Proc. of IEEE INFOCOM*, 2013, pp. 1079 – 1087.

[9] ——, "Convergence in Player-Specific Graphical Resource Allocation Games," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 11, pp. 2190–2199, 2012.

[10] Y. Schwartz, Y. Shavitt, and U. Weinsberg, "On the Diversity, Stability and Symmetry of End-to-End Internet Routes," in *Proc. of IEEE INFOCOM Workshop*, 2010, pp. 1–6.

[11] V. Bilò, A. Fanelli, M. Flammini, and L. Moscardelli, "Graphical Congestion Games." in *Proc. of WINE*, vol. 5385, 2008, pp. 70–81.

[12] L. S. Shapley, "A Value for n-Person Games," in *Contributions to the Theory of Games*, H. W. Kuhn and A. W. Tucker, Eds., 1953, pp. 307–317.

[13] "CAIDA. Automated Autonomous System (AS) ranking." [Online]. Available: http://as-rank.caida.org/data/

[14] D. J. A. Welsh and M. B. Powell, "An upper bound for the chromatic number of a graph and its application to timetabling problems," *The Computer Journal*, vol. 10, no. 1, pp. 85–86, 1967.

[15] E. Lloyd and S. Ramanathan, "On the complexity of distance-2 coloring," in *Proc. of International Conference on Computing and Information (ICCI)*, 1992, pp. 71–74.

[16] M. Korupolu and M. Dahlin, "Coordinated placement and replacement for large-scale distributed caches," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 6, pp. 1317–1329, 2002.

[17] Y. Chen, L. Qiu, W. Chen, L. Nguyen, and R. H. Katz, "Efficient and Adaptive Web Replication Using Content Clustering," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 6, pp. 979–994, 2003.

[18] S. Borst, V. Gupta, and A. Walid, "Distributed Caching Algorithms for Content Distribution Networks," in *Proc. of IEEE INFOCOM*, 2010, pp. 1478–1486.

[19] G. Pollatos, O. Telelis, and V. Zissimopoulos, "On the social cost of distributed selfish content replication," in *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, 2008, pp. 195–206.

[20] E. Jaho, M. Karaliopoulos, and I. Stavrakakis, "Social similarity favors cooperation: the distributed content replication case," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 3, pp. 601–613, 2013.

[21] B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, and J. Kubiatowicz, "Selfish caching in distributed systems: a game-theoretic analysis," in *Proc. of ACM symposium on Principles of Distributed Computing (PODC)*, 2004, pp. 21–30.

[22] G. Dán, "Cache-to-cache: Could ISPs cooperate to decrease peer-to-peer content distribution costs?" *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1469–1482, 2011.

[23] R. T. B. Ma, J. C. S. Lui, V. Misra, and D. Rubenstein, "Internet Economics: The Use of Shapley Value for ISP Settlement," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 775–787, 2010.

[24] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*, 2007.