Computer Networks 154 (2019) 1–11

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Revisiting core traffic growth in the presence of expanding CDNs

Francisco-Javier Moreno Muro[a,*], Nina Skorin-Kapov[b], Pablo Pavon-Marino[a,c]

[a] Universidad Politécnica de Cartagena, Cuartel de Antiguones, Plaza del Hospital 1, Cartagena 30202, Spain
[b] University Center of Defense, San Javier Air Force Base, MDE-UPCT, Santiago de la Ribera, Murcia, Spain
[c] E-lighthouse Network Solutions, Carlos III 42, 30203 Cartagena, Spain

A R T I C L E   I N F O

Article history:
Received 17 September 2018
Revised 18 January 2019
Accepted 4 March 2019
Available online 8 March 2019

Keywords:
CDN
Network Evolution
Traffic Growth

A B S T R A C T

Traffic growth forecasts announce a dramatic future for core networks, struggling to keep the pace of traffic augmentation. Internet traffic growth primarily stems from the proliferation of cloud services and the massive amounts of data distributed by the content delivery networks (CDNs) hosting these services. In this paper, we investigate the evolution of core traffic in the presence of growing CDNs. Expanding the capacities of existing data centers (DCs) directly translates the forecasted compound-annual-growth-rate (CAGR) of user traffic to the CAGR of carried core link traffic. On the other hand, expanding CDNs by building new geographically dispersed DCs can significantly reduce the predicted core traffic growth rates by placing content closer to the users. However, reducing DC-to-user traffic by building new DCs comes at a trade-off with increasing inter-DC content synchronization traffic. Thus, the resulting overall core traffic growth will depend on the types of services supported and their associated synchronization requirements. In this paper, we present a long-term evolution study to assess the implications of different CDN expansion strategies on core network traffic growth considering a mix of services in proportions and growth rates corresponding to well-known traffic forecasts. Our simulations indicate that CDNs may have significant incentive to build more DCs, depending on the service types they offer, and that current alarming traffic predictions may be somewhat overestimated in core networks in the presence of expanding CDNs.

1. Introduction

The new Internet paradigm, mainly based on cloud services and content delivery solutions, is leading to staggering traffic growth rates anticipating significant link congestion in future backbone networks [1–4]. To keep pace with these aggressive traffic forecasts and support the predicted IP traffic growth, numerous research efforts are underway in several fields. Newer technologies, such as space division multiplexing (SDM), are being researched as possible solutions [5], along with novel standards and algorithms to assess more effective ways to utilize the available resources to support future capacity demands. For instance, flexible or elastic optical networks have been proposed to achieve superior spectrum utilization and flexibility [6]. The main drivers fueling these research efforts are dramatic traffic growth forecasts. In this paper, we investigate the possible evolution of core traffic in the presence of growing content delivery networks (CDNs) to assess whether these predictions are fully justified.

CDNs consist of a set of data centers (DCs) interconnected by a set of links, either owned by the Content Service Provider or CDN, or leased from a telco core network. Typically, DCs are placed close to a core node to ease access to the network and content is replicated in a set of dispersed DCs allowing users to connect to the closest available DC hosting a replica of the desired content. As demands for cloud services increase, CDNs will be forced to expand to support this growth. CDN expansion can be realized by increasing existing DC capacities and/or opening new geographically dispersed DCs. While creating new DCs may imply higher capital expenditures (CAPEX) in comparison to expanding existing DCs, the resulting distribution can bring content closer to the users. This can in turn alleviate traffic in core links and reduce the mean round trip time. On the other hand, increasing the number of content replica locations for some applications may incur increased synchronization traffic limiting the overall core traffic reduction. The key contribution of this paper lies in evaluating the evolution of core traffic under different CDN expansion strate-

* Corresponding author.
E-mail address: javier.moreno@upct.es (F.-J.M. Muro).

https://doi.org/10.1016/j.comnet.2019.03.005

gies and assess the potential reduction in core traffic growth rates that can be achieved by adding new DC locations. To achieve this goal, we model a multi-period optimization problem considering various CDN expansion strategies, differing with respect to the rate of adding new DCs and criteria for choosing their potential locations. Furthermore, we consider a realistic mix of service types which determine the ratio of DC-to-user and inter-DC traffic for each service. The optimization problem searches for a DC placement over time as CDNs expand, as well as the associated replica placement and traffic routing. The objective is to minimize the total traffic in the core in each time period. The problem is solved by applying an iterative optimization algorithm in each time period using Integer Linear Programs for the replica placement and routing subproblems. Our results indicate that effective CDN expansion strategies could significantly alleviate future core congestion yielding traffic growths that are not as explosive as currently forecasted. However, the achievable benefits will depend on the types of cloud services offered which could play an important a role in future CDN growth decisions.

2. Related work

CDN optimization and DC placement for content distribution have been extensively studied in the literature. The work in [7] focuses on DC placement, exploring the trade-off between latency reduction and the resulting increase in inter-DC traffic. Minimizing the non-renewable energy consumption in datacenters is the key to solving the datacenter placement problem in [8]. The benefits of interconnecting CDNs between multiple content providers is analyzed in [9], where the benefits of collaboration between CDNs was shown to be advantageous. In [10], the authors present a framework for optimizing DC placement by taking into consideration the CAPEX and OPEX costs of the datacenter sites, latency, service availability and CO_2 emissions. Another important issue to treat in datacenter placement is resilience to disasters. The authors in [11] present a technique for disaster-aware DC placement and content management, along with a cost analysis of the placements based on the real-world cloud market.

Efficient replica placement and resource allocation in CDNs has been investigated widely in works such as [12] or [13] and references therein. For instance, in [12] the authors introduce joint manycast, anycast and a replica placement (MARP) strategies to find optimal and sub-optimal solutions for replica placement and the routing and wavelength assignment (RWA) of user-to-DC and inter-DC traffic. The anycast routing replica placement problem with the aim of minimizing the total resource usage in elastic optical networks is studied in [13]. In [14], the authors describe a Content-Centric Data Center Network (CCDN), a fully-distributed proposal for caching and content forwarding in order to improve performance with minimal cost. Minimizing the bandwidth and storage cost is the target for replica placement techniques in content distribution networks in [15,16]. The authors in [17] present a survey of several replica server algorithms for provisioning purposes considering traditional metrics, such as cost and QoS. In [18], the authors provide a survey of content placement algorithms and provisioning for Cloud-based CDNs taking into account dynamic changes in the popularity of content.

Regarding CDN traffic, some works are available in the literature studying DC-to-user traffic. For example, authors in [19] use five different 24-hours window demand traffic profiles in a multi-CDN interconnectivity environment. In [20], the study framework is an Information-Centric Networking (ICN) approach based on the CDN paradigm where it assumes a caching traffic model for two services (YouTube, web server) taking into account the popularity and the size of the content requested by users, focused on time availability for each content unit. However, these works do not consider inter-datacenter traffic for synchronization purposes. Due to the difficulty of obtaining DC-to-DC traffic information, [21] proposes an experimental framework to monitor real-time inter-datacenter traffic based on machine learning techniques tested in a university-wide network. The results show a high time fluctuation and high dependency on sampling accuracy. To the best of our knowledge, studies encompassing both user and synchronization traffic in real inter-DC networks are not publically available.

All the aforementioned works focus on the dynamics of CDNs and/or their associated replica/content placement strategies with the aim to optimize their dimensioning. In this work, however, we focus on the *impact* of such solutions on core traffic evolution. We believe this is the first study to consider the implications of growing CDNs, the nature of the services they offer and their associated routing paradigms on future traffic growth estimations in core networks.

3. Assumptions and problem model

In this work, we assume a carrier core network and a set of CDNs, such as Akamai, Level3, Amazon, etc. Each CDN is composed of a set of DCs interconnected by links leased from the core network. CDNs that own their own network infrastructures do not affect core traffic and are, thus, not considered in this study. As customary, we assume the DCs are located in the proximity of core nodes giving them direct access to the carrier network. Content providers offer services in the form of applications, such as Instagram, YouTube, and Netflix, which are served by one or more CDNs. In some cases, such as Facebook or Google, the content provider is also the owner of the CDN used [22]. In other cases, the content provider offers its services via one or multiple CDNs owned by different vendors. For example, Netflix uses three different CDNs (Akamai, Limelight, Level3) to serve their content to end users and an additional CDN (Amazon) for monitoring and control [23].

3.1. Core traffic model

The traffic in the core network can be categorized into 3 main categories. The first is user-to-user traffic which is independent of cloud services and is mainly handled by carrier networks. We assume this traffic type to be bidirectional, proportional to the product of the end nodes population, and ruled by a fixed compound annual growth rate (CAGR). The latter two, DC-to-user and DC-to-DC traffic, constitute core traffic generated by cloud-based services [1,2] and are the main focus of this work. DC-to-user traffic represents end-user-driven communication where users access cloud content and services, typically from the closest DC hosting a content replica. DC-to-DC traffic, on the other hand, encompasses inter-DC traffic which is related to content replication, monitoring and synchronization/updating between DCs.

We consider a set of applications which generate cloud-based traffic, such as Facebook, YouTube, Netflix, Xbox Games, iTunes, etc., each offering a specific service. The set of services considered, as given in [2], includes long Internet video (such as that offered by Netflix or Hulu); short Internet video (such as YouTube); web, email and data (which includes web, email, instant messaging and other data traffic excluding file sharing); file sharing (P2P and other file transfers) and finally online gaming. Although, in general, an application can offer several services, for simplicity we only consider one service per application. The content of each application is replicated in a subset of DCs in one or more CDNs serving that application. Each service has a specific CAGR value, according to [2], which is considered to be the CAGR of the traffic associated with each application offering that service.

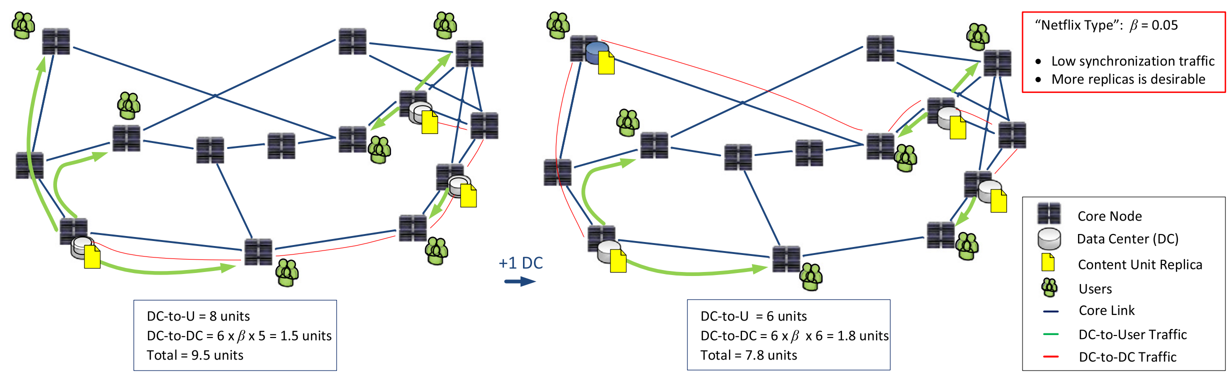

Fig. 1. An example of CDN expansion and the associated replica placement for low-β services.

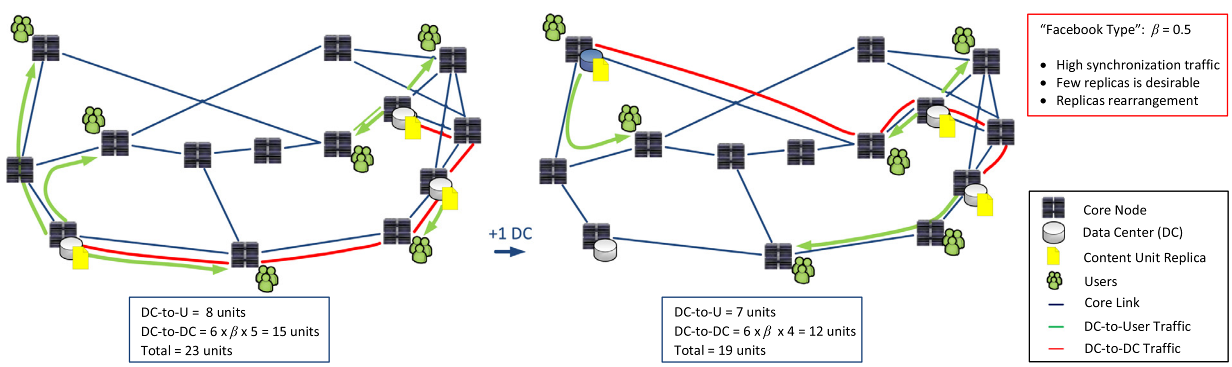

Fig. 2. An example of CDN expansion and the associated replica placement for high-β services.

As opposed to user-to-user traffic modeled by a classical traffic matrix, where both the source and destination nodes are known, in cloud-based DC-to-user traffic, user demands can be served by one of many DCs, typically the closest one hosting a content replica. This means that only one end node (that associated with the user) is known while the other can be any one within a given subset of DCs. This implies the anycast routing paradigm where a source node can connect to any node within a given subset of destination nodes. We assume that the total traffic associated with each application for users covered by a specific core node is proportional to the population of the biggest city where the node is placed. Furthermore, the content of each application is split into 100 content units, and the weight of each unit is proportional to its popularity modeled as a $1/f$ Zipf distribution where the most popular content unit has 100 times more weight than the 100-th most popular one. Consequently, the most popular content unit of the most populated node has the highest traffic weight while the least popular content unit in the node with the lowest population has the lowest traffic weight. As each application is served by one or more CDNs, the total traffic associated with each application in each user node is divided proportionally between the CDNs serving it. Furthermore, the content units associated with an application are replicated in at least two DCs in each CDN serving that application.

Content replica migration and synchronization between DCs comprise the second type of cloud-based traffic, DC-to-DC traffic, which assumes multicast communication between the DCs of each CDN hosting replicas of a given content unit. If the replica placement is unknown, this translates to the manycast routing paradigm which interconnects only a subset of nodes (i.e., DCs) from the possible destination set.

To tune the ratio of DC-to-user and DC-to-DC traffic, we associate a factor $\beta \in [0.05, 0.5]$ with each service. For example, a value $\beta = 0.1$ implies that if a content is accessed by the user summing a flow of 10 Gbps, the content update flow to keep replicas synchronized is of 1 Gbps. The value of β of an application depends on the nature of the services that it offers. For instance, long internet video applications, such as Netflix, have low β values where the DC-to-DC traffic required to update a video is small in comparison to the DC-to-user traffic generated by a large number of users accessing that content. On the other hand, social media applications, such as Instagram or Facebook, have higher β values where users constantly upload and modify content which must be updated and synchronized between multiple DCs yielding more significant DC-to-DC traffic.

3.2. CDN growth policies

While growth rates of classical user-to-user traffic following standard unicast shortest path routing directly translate to carried traffic growth in the links, cloud-based traffic growth has a different effect due the alternative routing paradigm implied and their dependency on DC distribution. Recall that user demands are served from any DC hosting a replica of the desired content and inter-DC traffic involves multicast communication between the set of DCs hosting the replicas. Consequently, the induced cloud-based traffic in the core is very dependent on the number and location of the content replicas and any changes therein.

Here, we investigate the effects of adding new geographically dispersed DCs enabling enhanced replica placement, which can potentially reduce the overall traffic growth in the core over time. Figs. 1 and 2 illustrate this concept for two different service types offered by a CDN. In Fig. 1, we assume a low β service ($\beta = 0.05$) which does not imply significant synchronization traffic compared to the anycast user traffic incurred. The CDN is initially comprised of three available DCs (shown on the left) with a replica of the

Algorithm 1: Algorithm MP-CTE

```
Input:
    Set N  //Set of nodes in the core;
    Set U  //Set of content units;
    Set A  //Set of applications;
    Set C  //Set of CDNs to serve the apps traffic;
    Set D_c  // Initial set of DCs in each CDN c ∈ C;
    H_u2u(0)  //Offered u2u traffic in time period 0;
    H_ac(0)  //Offered CDN traffic per application a served by CDN c in time period 0;
    β_a  //Beta factor per application a;
    CAGR_u2u, CAGR_a  //CAGR for u2u traffic and for CDN traffic per application a;
    G  //CDN growth factor;
    T  //Number of time periods;
Output:
    coreTraffic(t), t = 0...T  //Total traffic in the core in each time period t;
Begin:
Calculate R_u2u //Set of shortest paths for u2u routing, constant ∀ time periods;
foreach c ∈ C do
    Execute ILP-RP for replica placement for all content units u served by CDN c time period 0;
    Calculate anycast routes R_u2dc(c,0) as shortest path to closest replica for CDN c in time period 0;
    Execute ILP-MC to calculate multicast trees R_dc2dc(c,0) for CDN c in time period 0;
end
R(0)={R_u2u , ∪_{c∈C} R_u2dc(c,0), ∪_{c∈C} R_dc2dc(c,0)} //Complete routing solution in time period 0;
Calculate coreTraffic(0)  //Total traffic in the links based on offered traffic and routing solution R(0);
t = 1 //Current time period;
t'_c = 0, ∀ c //Last time period a new DC was added to CDN c;
do
    H_u2u(t) = H_u2u(0) · (1 + CAGR_u2u)^t  //Calculate u2u traffic in time period t;
    R(t)=R(t − 1) //Initially assume existing routing solution;
    foreach c ∈ C do
        Calculate H_c(t) //Total offered traffic for CDN c in time period t, where offered traffic per
          application grows as H_ac(t)= H_ac(0) · (1 + CAGR_a)^t ;
        Calculate coreTraffic(c,t) for CDN c with existing DCs and routing solution;
        addDC = G · (H_c(t) − H_c(t'_c))/H_c(t'_c);
        if addDC ≥ 1 then
            newDC = 0;
            foreach n ∈ N/D_c do
                Execute ILP-RP for replica placement for all content units u served by CDN c time
                  period t assuming a new DC is placed at node n;
                Calculate anycast routes R_u2dc(c,0) as shortest path to closest replica;
                Execute ILP-MC to calculate resulting multicast trees R_dc2dc(c,t,n);
                Calculate coreTraffic(c,t,n);
                if coreTraffic(c,t,n) < coreTraffic(c,t) then
                    newDC = n;
                end
            end
            if newDC ≠ 0 then
                D_c = D_c ∪ newDC;
                Update R(t) with R_u2dc(c,t,newDC) and R_dc2dc(c,t,newDC);
                t'_c = t;
            end
        end
    end
    Calculate coreTraffic(t)  //Total traffic in the core in time period t;
    t++;
while t ≤ T;
End;
```

Fig. 3. Pseudocode of the MP-CTE optimization algorithm.

considered content placed in each of them and a set of users demanding this content in six different nodes. In this scenario, the DC-to-user traffic (represented by green lines) requires 8 hops to satisfy the user demands, while the multicast tree (in red), used to deliver and synchronize replicas among the DCs, has a size of 5 hops. If each user demand requires 1 traffic unit, this would result in 8 units of DC-to-user traffic over all links. Then for each of the 6 users, the associated synchronization traffic would be $\beta x 5$, where 5 is the size of the multicast tree. Thus, the overall synchronization traffic would equal $6x \quad \beta x \; 5 = 1.5$ units, yielding a total of 9.5 units covering both traffic types. Note, the thickness of the lines of the DC-to-user and DC-to-DC traffic illustrates their relative weight based on β and the total traffic in the links is a combination of these two traffic types.

If the CDN owner decides to create a new DC (shown on the right), a new replica could be placed in the new DC providing an improvement of 2 hops in total DC-to-user traffic routing, but at the cost of augmenting the multicast tree by one hop. Since the multicast traffic for low β services is much lower compared to the anycast traffic, this is an advantageous trade-off. The DC-to-user traffic reduces to 6 units, while the DC-to-DC traffic increases to 1.8, giving a reduction in the overall traffic to 7.8 units. Typically, increasing the number of replicas as new DCs are added is a favorable replica placement strategy for services with low β values.

Fig. 2, shows the result of replica placement under the same conditions but applied to a high β service which requires significant synchronization traffic. We can see that the DC-to-DC traffic is much higher (15 units), giving an overall traffic of 23 units. In this case, the addition of a new DC is optimally leveraged by rearranging the replicas (without augmenting their number) to obtain a more beneficial replica placement. We can see that the anycast traffic is only reduced by 1 hop, compared to 2 hops in the previous case, but the multicast traffic is not incremented (in fact it is even reduced in this example) which provides a superior overall solution of 19 traffic units. In both cases, the addition of a new DC can allow for enhanced replica placement for global traffic reduction and could motivate future CDN expansion decisions.

To assess the potential advantages of this approach, we consider a multi-period planning scheme with various CDN growth policies as follows. We assume that in year zero all the available CDNs have an initial set of DCs placed in the proximities of core nodes. Since DC location decisions may depend in several factors beyond the scope of this work, including political or geographical, the initial DC placement is random. As cloud-based traffic increases according to the aforementioned traffic models, the CDNs serving this traffic must grow to support it, either by adding capacity to existing DCs and/or opening new DC locations. To model different CDN growth policies, we define a growth factor G as follows. For a value of $G=0$, traffic growth is supported exclusively by increasing the capacity of current data centers, i.e., the CDN never increases the number of existing DCs. For $G=1$, an additional DC can be built if the traffic carried by the CDN has doubled from the last year a new DC was opened. If the CDN growth factor takes on a value of $G=2$, a new DC can be created if the traffic has increased by 50% with respect to the last time the CDN was expanded. Note that, if there is no resource benefit of adding a new DC, it is not built, i.e. the CDN expansion policy offers the possibility of adding new DCs but it is not compulsory.

The decision on where to place a new datacenter can depend on several factors, such as geographic diversity (to improving redundancy and DC-to-user latency), local factors (e.g. power concessions, taxes, available workforce, etc.) and network costs (transmission costs) [10]. Additionally, risk factors, such as susceptibility to natural disasters or political instability, can play a role in DC placement [11]. Since the main objective of this paper is to investigate the effects of CDN expansion on the total core traffic in the links (which comprises the main network cost) we consider the following two DC placement strategies. The first, denoted as *TrOpt*, optimally tunes the trade-off between the anycast profit and the manycast cost of the DC-to-user and DC-to-DC traffic, respectively, served by each CDN. Recall that increasing the number of geographically distributed DCs can reduce end-user-driven traffic requirements by allowing users to access a closer DC, but may incur increased inter-DC content replication and synchronization traffic. This strategy performs joint DC and replica placement to minimize the total traffic, i.e. the best-case scenario. Since it is not realistic to assume that only network costs determine DC placement, we also consider a second strategy, denoted as *Rand*, which selects a location at random among the subset of available locations that reduce the overall traffic. This later approach is aimed to simulate the case where a combination of alternative factors and constraints guide the placement process.

3.3. Problem definition

Given a core network, composed of a set of nodes, N, a set of links, E, and a set of CDNs, C. Each CDN is composed of a set of initial DCs D_c, where D_c is a subset of nodes from N. Given is also a set A of applications, each associated with a subset of CDNs which serve that application. Each application is composed of a set U of content units and associated with a specific β_a value depending on the service type it offers. Provided is also the initial offered traffic as follows. Non-CDN traffic is given as a traffic matrix describing the total traffic demands between node pairs (*H_u2u(0)*). CDN traffic, on the other hand, is given per application and is associated only to the user nodes as the DC node(s) which will serve each user are unknown ($H_{ac}(0)$). The traffic demand for each application is then split for each content unit according to Zip's law where the n^{th} content unit has associated n times more traffic than the first content unit. The traffic associated with each content unit is further split uniformly between the CDNs serving that application. The annual traffic growth rates (CAGRs) are given for the total user-to-user traffic (*CAGR_u2u*) and for traffic per application ($CAGR_a$). Finally, factor G and the DC placement strategy (*Rand*/*TrOpt*) of the CDN expansion policy considered are also provided as input parameters.

To assess the evolution of core traffic under different CDN expansion policies, we solve the following optimization problem for T consecutive time periods, i.e., years. The problem searches for a DC and replica placement and traffic routing with the objective to minimize the overall traffic in the links. Note, we solve the optimization problem for each time period independently, using as the offered traffic and current DCs of each CDN of the current time period as input to the next time period.

The decision variables are as follows: whether a new DC will be added for each CDN in that time period and its location (updated sets D_c); the number of replicas of each content unit of each application in the CDNs serving that application and their locations; the user-to-user traffic routing; the anycast routes connecting the traffic of each content unit of each application from each user node to a replica in the CDN serving that traffic; the multicast trees interconnecting the replicas of each content unit of each application in each CDN.

The problem is constrained by the CDN expansion policy described in the previous subsection, where new DCs can be added according to factor G and their location is chosen based on the *Rand*/*TrOpt* placement policy. Furthermore, each content unit of each application must have a minimum of r replicas in each CDN serving that application. Replicas of each content unit must be placed at feasible DC locations, according to updated sets D_c. Links are assumed bidirectional and unconstrained, i.e., are assumed to be expanded by the carriers on demand, to avoid blocking and asses the maximal potential reduction of traffic growth rate achievable via CDN expansion. Note that this approach is consistent with conventional carrier methodology, where link capacities are upgraded once or twice a year, to adjust them to the forecasted traffic and thus avoiding traffic blocking. Similarly, there is no constraint on the capacity of existing DCs.

4. The multiperiod core traffic evolution (MP-CTE) algorithm

The problem proposed in Subsection 3.3., which tackles the replica and datacenter placement, and anycast and manycast routing, over time is a complex optimization problem, that is solved using a so-called math-heuristic technique, with a heuristic scheme that sequentially solves the problem in each time period, and iteratively applies ILPs as subroutines for smaller parts of the problem within each time period. Fig. 3 depicts the pseudocode of the proposed Multi-Period Core Traffic Evolution (MP-CTE) algorithm.

User-to-user traffic is routed on the shortest path (in terms of the number of hops) between end nodes (line 1). Since link capacities are unconstrained, the user-to-user traffic routing remains unchanged over time and does not affect the optimization. Given a DC placement for each CDN in time period 0, the number and placement of replicas for all content units served by that CDN is calculated by running an Integer Linear Program (ILP), called ILP-RP, aimed at minimizing the estimated total CDN traffic (DC-to-user and DC-to-DC) in the core links (line 3). The proposed ILP is described in the next subsection. Based on the calculated replica placement, the anycast routes are extracted using shortest path routing to the closest replica (line 4), while the multicast routes are solved by running an exact ILP, referred to as ILP-MC and described in Subsection 4.2 (line 5). The traffic in the core is then calculated based on the offered traffic and the user-to-user, DC-to-user and DC-to-DC routing solutions (lines 7–8).

The pseudocode in Fig. 3 illustrates the case of *TrOpt* placement aimed to minimize the total CDN bandwidth consumption in the core (lines 20–28). For each potential location, the number and placement of replicas for all content units served by that CDN is recalculated with ILP-RP, making use of all existing CDN DCs and the new tentative DC. Anycast and multicast routing is then solved and resulting core traffic is calculated. The node giving the minimum traffic is selected as the new DC location and the routing solution is updated accordingly (lines 29–33). If no candidate DC location improves the routing solution in terms of the resulting core traffic, we assume that the CDN has no incentive to build the new DC and continues serving the users with its current DC footprint. In the case of *Rand* placement, a single node is chosen at random among the potential DC locations, substituting lines 20–28, and is accepted only if it improves the current solution. Finally, the total core traffic for each time period is calculated for the full routing solution and offered traffic over all CDNs (line 36).

This proposed comprehensive multi-period approach allows us to evaluate the effects of different CDN growth policies on overall core traffic growth rates, as well as the effects of the type of applications (services) served based on different ratios of DC-to-user and DC-to-DC traffic (factor β) modifying the associated anycast profit/manycast cost trade-off.

4.1. ILP for replica placement (ILP-RP)

In this subsection, we present the ILP used to solve replica placement in a single CDN with a given DC placement, referred to as ILP-RP. The main objective is to minimize the total traffic traversing the core network considering anycast routing for the user-to-DC traffic and a lower bound on the multicast routing required for the inter-DC synchronization traffic for each content unit. The multicast subproblem is not solved exactly due to scalability issues as this ILP subroutine is called for each CDN and content unit in each time period for each potential location. However, note that once a replica placement is chosen, multicast routing for DC-to-DC traffic is solved exactly by applying ILP-MC (see next subsection). The input traffic follows the traffic model described in Section 3.1 where factor β tunes the ratio of DC-to-user and DC-to-DC traffic.

The set of input parameters required by the ILP algorithm are as follows:

- $\mathcal{U}$: Set of content units.
- $\mathcal{A}$: Set of applications in a CDN.
- $\mathcal{N}$: Set of nodes in the core network.
- $\mathcal{D}$: Set of datacenters of a CDN.
- $\hat{Z}_u, u \in \mathcal{U}$: Popularity of a content unit u represented by a normalized Zipf distribution.
- $h_a, a \in \mathcal{A}$: Total offered traffic of the application a.
- $\hat{p}_n, n \in \mathcal{N}$: Population of a node n normalized to the total population.
- $C_{nd}, n \in \mathcal{N}, d \in \mathcal{D}$: Cost in number of hops of the shortest path between a user node n and a datacenter d.
- $\beta_a, a \in \mathcal{A}$: β factor of an application a.
- r: minimum number of replicas of each content unit.

The decision variables which comprise the solution of the ILP are as follows:

- Cl_{uand}, $u \in \mathcal{U}$, $a \in \mathcal{A}$, $n \in \mathcal{N}$, $d \in \mathcal{D}$: 1 if for a content unit u belonging an application a, the datacenter d is the closest one with the content u to a user node n, 0 otherwise.
- $r_{uad}, u \in \mathcal{U}, a \in \mathcal{A}, d \in \mathcal{D}$: 1 if the datacenter d has a replica of the content unit u belonging to an application a, 0 otherwise.

The objective function is:

$$\sum_{u,a,n,d} \hat{z}_u \cdot h_a \cdot \hat{p}_n \cdot c_{nd} \cdot cl_{uand} + \sum_{u,a,d} \hat{z}_u \cdot h_a \cdot \beta_a \cdot \left(\sum_{u,\,a,\,d} r_{uad} - 1 \right) \text{subject to :} \tag{1}$$

Eq. (1) minimizes the total core traffic in the links by balancing the two aforementioned traffic types. The left part minimizes the traffic served to satisfy the anycast needs, while the right part offers a simple lower bound on the multicast synchronizing traffic. The lower bound on the minimum number of hops in the associated multicast tree is calculated as the number of replicas minus 1 by assuming direct links between all datacenters hosting a replica.

The constraints are given below:

$$\sum_{d \in \mathcal{D}} cl_{uand} = 1 \; \forall \, u \in \mathcal{U}, \;\; a \in \mathcal{A}, \;\; n \in \mathcal{N} \tag{2}$$

$$\sum_{d \in \mathcal{D}} r_{uad} \geq r \; \forall \, u \in \mathcal{U}, \;\; a \in \mathcal{A} \tag{3}$$

$$\sum_{u \in \mathcal{U}} \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}} cl_{uand} \leq r_{uad} \; \forall \, n \in \mathcal{N} \tag{4}$$

Constraint (2) ensures that only one datacenter is denoted as closest to a node n for each content unit u belonging to an application a. Eq. (3) requires that at least r replicas for each content unit are placed in the CDN. Finally, Eq. (4) ensures that the number of closest datacenters to a node n cannot be greater than the total number of replicas.

4.2. ILP for multicast routing (ILP-MC)

In this subsection, we present the ILP-MC formulation used to solve the multicast routing problem for replica synchronization. The objective function aims to minimize the number of links used to build the multicast tree, that is also minimizing the total bandwidth consumed. ILP-MC is a variation of a formulation for multiple Steiner trees published in [24], and the code of ILP-MC is open and available at [25].

The set of input parameters needed in ILP-MC are as follows:

- ε: Set of links in the core network.
- $\mathcal{N}$: Set of nodes in the core network.
- $\mathcal{D}_u$: Set of datacenters of a CDN with a replica of the content unit u.
- c_e: cost of the link e.
- o: origin of the tree.

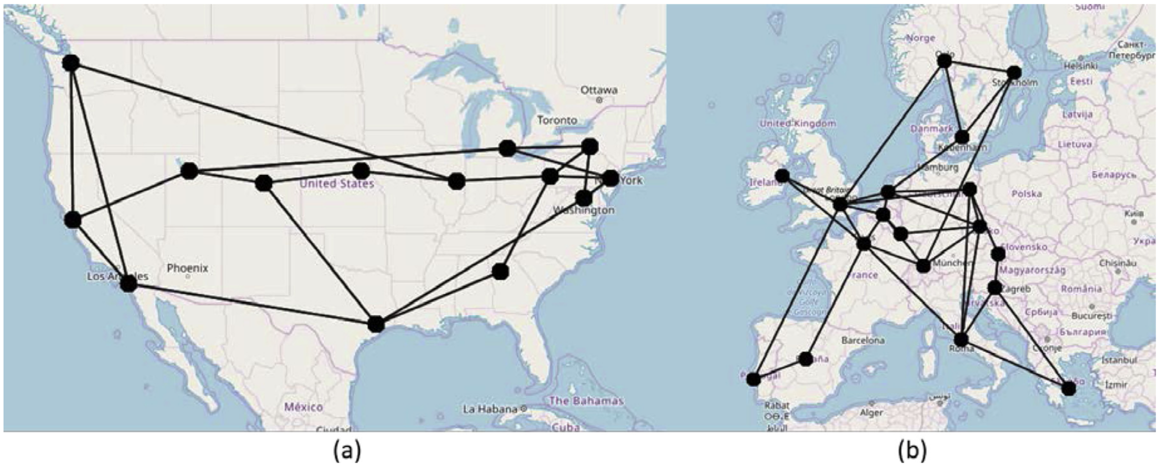

Fig. 4. The (a) NSFNet and (b) EON topologies.

The decisions variable to retrieve the solutions of the ILP are:

- $x_{e:}$ 1 if a link e is in the tree, 0 otherwise.
- $x_{ed:}$ 1 if a link e is in the tree subpath from origin to datacenter d, 0 otherwise.

The objective function is:

$$\sum_{e} c_e \cdot x_e \text{ subject to :} \qquad (5)$$

Eq. (5) minimizes the total cost in the multicast tree with the minimum number of links (hops) used in the tree. In this particular case, we consider equal link cost equal to 1.

The objective function is constrained by:

$$x_{ed} \le x_e, \quad \forall\, e \in \mathcal{E},\ d \in \mathcal{D}_u \qquad (6)$$

$$\sum_{e\in\delta^+(n)} x_{ed} - \sum_{e\in\delta^-(n)} x_{ed} \begin{cases} = 1,\ if\ n = o \\ = -1,\ if\ n = d, \forall\, d \in \mathcal{D}_u,\ n \in \mathcal{N} \\ = 0\ otherwise \end{cases} \qquad (7)$$

Eq. (6) ensures that a link e belongs to the tree if it appears in any path to the datacenter (destination) d. The expression shown in (7) is the flow conservation constraint for the individual path of each multicast tree, where $e \in \delta^+(n)$ are the incoming links to a node n, and $e \in \delta^-(n)$ the outgoing links from a node n.

5. Simulation set up and numerical results

This section collects and analyzes the results of a core traffic study aimed at evaluating the future traffic growth for different CDN growth policies on two reference topologies and a parametric study to assess the influence of the nature of applications (services) on this evolution. Both simulations were implemented in Net2Plan [25,26], a network design and planning tool which includes the JOM library [27] (a Java-based interface for the CPLEX solver) used to execute the ILP algorithm.

5.1. Traffic forecasts under varying CDN growth policies

The multi-period simulation of core network evolution was run on the 14-node NSFNet and an 18-node European Optical Network (EON), available in [25] and shown in Fig. 4 which provides information about the node populations, for a period of 15 years from 2017. The core traffic was modeled in accordance with the assumptions described in Section 3.1, where user-to-user traffic comprised 24% of the total traffic demands with a CAGR of 14%, while the remaining 76% of Internet user traffic represented DC-to-user and DC-to-DC traffic. Since we are interested in assessing relative core traffic growths, the total user offered traffic value in year zero was normalized to one unit. The cloud-based traffic was assumed to be generated by 20 different applications, each offering a single service. The types of services, CAGR rates per service and user traffic distribution are shown in Table 1 and follow the Cisco VNI [2] and Sandvine [3] traffic reports. Representative applications and arbitrary β values are also summarized in Table 1.

For simplicity, all applications offering the same service type were considered to have the same β value. The total cloud-based traffic was served by 5 CDNs, each initially comprised of 3 DCs randomly placed and each application was served by 1 to 3 different CDNs. The content units of each application were required to have at least 2 replicas in each CDN serving it. The simulation was run for 3 different CDN growth factors $G = \{0, 1, 2\}$ and 20 executions per each G value. For growth factors $G = 1$ and 2, the two different DC placement strategies described by the Problem Model were applied: *Rand* and *TrOpt*. The maximum execution time of the ILP solver was set to 60 s. In the large majority of instances, the optimum placement was found before this time limit.

The total normalized traffic in the links over a 15-year time period for the NSFNet and EON reference topologies is shown in Figs. 5(a) and (b), respectively, expressed on a semi-logarithmic scale on the vertical axis. We can see that by adding new DCs (growth strategies with values $G = 1$ and 2), the total carried traffic in the links over time is significantly reduced in comparison to CDN growth supported solely by expanding existing capacities ($G = 0$). This reduction is especially dramatic for the *TrOpt* placement policy which optimizes the location of the newly added DCs to reduce the overall traffic. Note, for a value of $G = 0$, the traffic trend follows a straight line fitting exponential curves with CAGR values of 27.4% for the NSFNet and 27.2% for the EON network. This is in line with the Cisco VNI traffic forecasts of 27% [2] and indicates that the user traffic growth directly translates to the total traffic growth in the links. For $G = 1$ and 2, we can see that the total link traffic growth rate is not constant. During the initial growth period (2–8 years), it is significantly reduced (particularly for $G = 2$) up to a saturation point. After this point, the traffic growth rate is no longer dependent on the CDN growth policy. This occurs due to the limited size of the network topologies, where adding new DCs is no longer beneficial since all users can access neighboring DCs. Thus, for long term planning, careful DC placement (*Rand*/*TrOpt*) is more definitive than the rate of adding new DCs (G). Note that the

Table 1
Internet service features.

Service	Representative application	Portion of user traffic	CAGR	β
Long Internet Video	Netflix	0.465	31%	0.05
Short Internet Video	YouTube	0.252	31%	0.25
Web, email, data	Instagram	0.154	17%	0.5
File sharing	Bit Torrent	0.113	0%	0.45
Online Gaming	Xbox Live	0.013	62%	0.05

Table 2
Traffic CAGR for the 15 year period (and carried traffic after 15 years).

CDN Growth Policy	$G=0$		$G=1$				$G=2$			
			Rand		*TrOpt*		$G=2$		$G=2$	
Network	NSFNet	EON	NSFNet	EON	NSFNet	EON	NSFNet	EON	NSFNet	EON
Total traffic in links	**27.4% (59.4)**	**27.2% (55.9)**	**23.6% (42.2)**	**24.4% (44.6)**	**19.5% (26.5)**	**23.2% (38.7)**	**23.6% (42.5)**	**24.3% (44.2)**	**19.5% (27.1)**	**23.0% (38.1)**
User-to-user traffic	14% (2.3)	14% (2.7)	14% (2.3)	14% (2.7)	14% (2.3)	14% (2.7)	14% (2.3)	14% (2.7)	14% (2.3)	14% (2.7)
DC-to-user traffic	30.5% (46.0)	30.8% (41.1)	25.3% (27.0)	27.9% (31.1)	18.1% (14.0)	23.9% (22.0)	25.2% (26.8)	26.6% (30.8)	16.7% (12.3)	22.9% (20.3)
DC-to-DC traffic	24.9% (10.9)	24.9% (12.1)	28.2% (12.4)	23.6% (10.8)	26.4% (11.2)	28.4% (14.0)	28.4% (12.5)	23.6% (10.8)	28.4% (12.5)	29.4% (15.8)

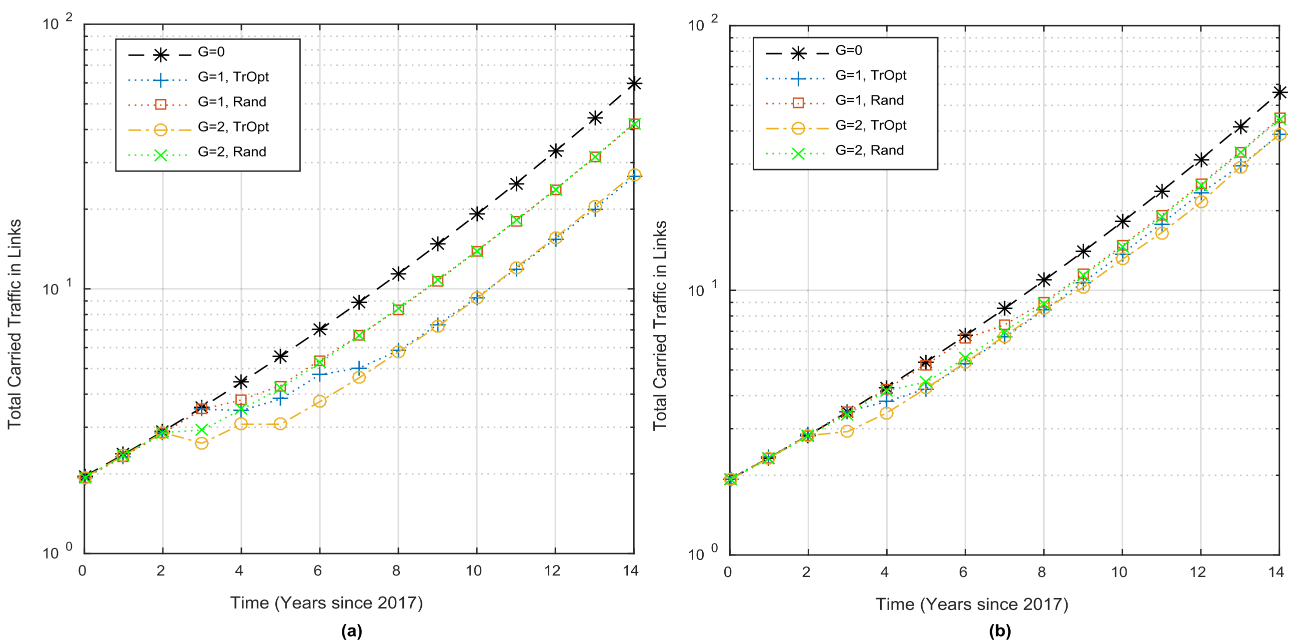

Fig. 5. The total traffic in the core links for the (a) NSFNet and (b) EON topologies over a 15-year period.

achieved reduction in traffic growth is more significant in the NSF network than in the EON network. This stems from the differences in the connectivity of the considered topologies. Namely, the EON has a diameter of 4 hops while the average path length between node pairs is 2.24 hops, in comparison to 3 and 2.14 hops, respectively, for the NSF network. This allows for shorter paths between users and DC locations from year zero, thus limiting the potential benefit achieved by adding more DCs.

For further analysis, Table 2 gives the CAGR values for the different growth policies for the 15 year period. Furthermore, the normalized value of the global carried traffic after 15 years is shown (in traffic units), where the user offered traffic in year zero is considered to be 1 unit. We can see that for $G=1$ and a *Rand* DC placement, the annual growth rate of the total traffic is reduced from 27.4% to 23.6% in the NSFNet and from 27.2% to 24.4% in the EON network. This means the carried traffic after 15 years is reduced from 59.4 to 42.2 units (NSF) and 55.9 to 44.6 units (EON) corresponding to reductions of 29% and 20%, respectively. By employing the *TrOpt* DC placement strategy and $G=1$, the annual growth rate of the total traffic is reduced to 19.5% in the NSFNet and 23.0% in the EON network. This corresponds to traffic reductions of 56% (NSF) and 36% (EON) after 15 years when compared to $G=0$. Results are similar for $G=2$. These staggering savings indicate that CDN expansion (in terms of the number and location of DCs) can be a very effective mechanism for reducing future core traffic growth.

Table 2 also indicates the traffic breakdown between the 3 traffic types. Naturally, user-to-user traffic is not affected by CDN expansion policies while the cloud-based traffic types exhibit a clear dependency. As expected, DC-to-user is significantly reduced when CDNs grow in terms of the number of DC locations ($G=1$ and 2), particularly for the *TrOpt* placement strategy, in comparison to growth realized by expanding existing facilities ($G=0$). On the other hand, DC-to-DC traffic typically grows as CDNs add new DC locations (except for the case of *Rand* placement in the EON network), but this growth is less pronounced due to flexible replica placement. Note, adding new DC locations does not necessarily imply an increase in the number of replicas.

Table 3
Average execution times for the MP-CTE algorithm (and sub-routine ILP-RP) for 20 instances of $C = 5$, $D = 3$, $A = 20$ AND *TROPT*.

Network	$G = 0$	$G = 1$	$G = 2$
NSFNet	5,2 s (1,0 s)	55,1 s (1,1 s)	67,9 s (1,2 s)
EON	6,5 s (1,3 s)	70,1 s (1,4 s)	133,1 s (2,1 s)

Table 4
Traffic CAGR for the 15 year period (and carried traffic after 15 years) for the NSFNET.

CDN Growth Policy	*A* = 10				*A* = 10			
	C = 5		*C* = 5		*C* = 5		*C* = 5	
G	*D* = 2	*D* = 3	*D* = 2	*D* = 3	*D* = 2	*D* = 3	*D* = 2	*D* = 3
0	27.0% (57,3)	26.9% (57.1)	27.0% (54,7)	26.7% (49,5)	27,4% (61,2)	27,4% (59,4)	27,2% (57,0)	26,8% (49,4)
1	18.9% (24,3)	18.9% (24.5)	19,2% (23,8)	19,7% (23,6)	19,5% (26,6)	19,5% (26,6)	19,9% (26,0)	20,7% (25,6)
2	18.7% (24,2)	19,3% (25,4)	19,1% (24.6)	20,2% (25.0)	19,5% (27,2)	19,5% (27,1)	19,8% (26,4)	20,7% (26,6)

Table 5
Traffic CAGR for the 15 year period (and carried traffic after 15 years) for the EON network.

CDN Growth Policy	*A* = 10				*A* = 10			
	C = 5		*C* = 5		*C* = 5		*C* = 5	
G	*D* = 2	*D* = 3	*D* = 2	*D* = 3	*D* = 2	*D* = 3	*D* = 2	*D* = 3
0	27.5% (60,2)	26.9% (57.6)	27.5% (61,4)	26.8% (56,8)	27,2% (59,5)	27,2% (55,9)	27,3% (60,0)	26,9% (57,8)
1	23.0% (36,7)	22.9% (36.5)	22,7% (36,6)	19,7% (36,2)	23,1% (38,5)	23,2% (38,7)	23,0% (38,1)	23,0% (37,9)
2	22.6% (35,8)	22,6% (36,5)	22,1% (34.7)	22,4% (35.7)	23,0% (37,8)	23,0% (38,1)	22,4% (22,7)	22,6% (37,0)

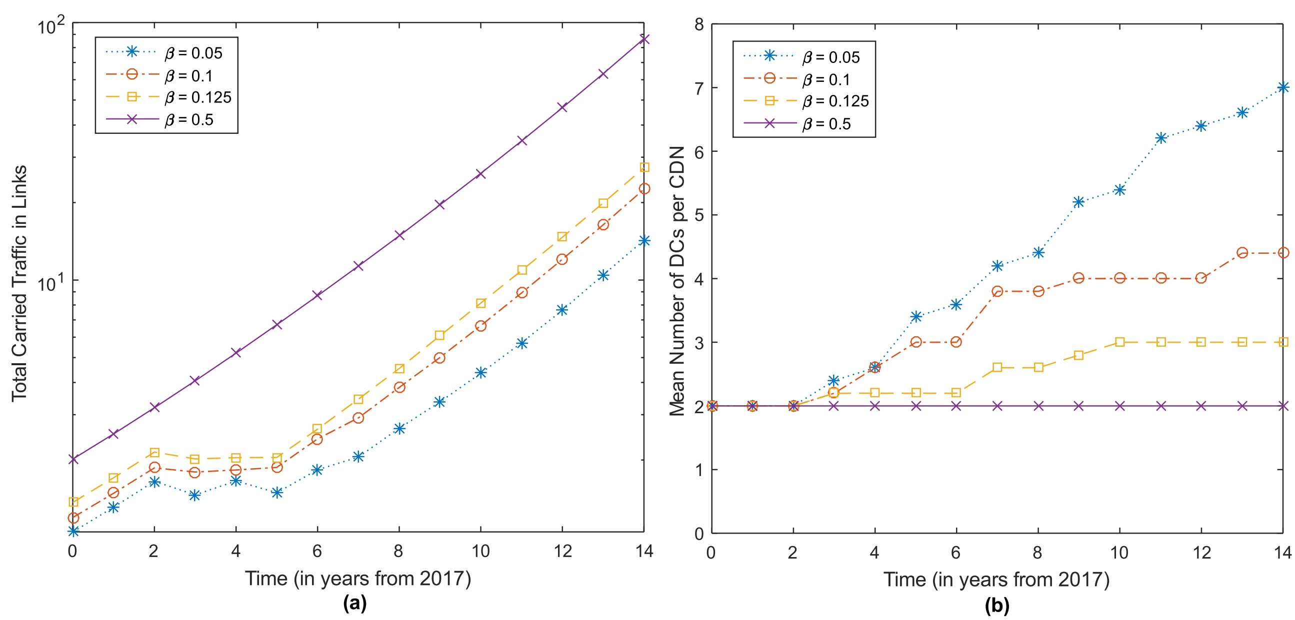

Fig. 6. The (a) total carried traffic in the links and the (b) number of DCs per CDN for different applications (i.e., service-types) for the NSF network.

Regarding the execution time of the algorithm, it is dominated by the ILP-RP subroutine which is executed for each CDN initially and each time a new DC can potentially be added for each potential location. While the maximal execution time of the ILP solver was set to 60 s, in most cases the optimal solution for ILP-RP was found in under 2 s, and all instances of the MP-CTE algorithm ran under 3 min. Table 3 gives the average execution times (in seconds) for the MP-CTE algorithm the two reference topologies using *TrOpt* growth strategy (which is more complex than *Rand*), varying *G* in a set of 20 executions per configuration, as well as the average execution time for a single instance of the ILP-RP subroutines in parenthesis.

5.2. Traffic forecasts under a varying number of applications, DCs and CDNs

To gain further insight into our results, we ran the MP-CTE algorithm for a varying number of applications ($A = 10$ and 20), CDNs ($C = 5$ and 10) and initial DCs ($D = 2$ and 3) for the *TrOpt* strategy for both networks and growths factors $G = 0$, 1 and 2. The results are shown in Tables 4 and 5 for the NSFNet and EON, respectively. We can see that the traffic reduction of the normalized traffic is fairly insensitive to the number of applications and CDNs. The initial number of DCs plays a small role, where a higher initial number of DCs naturally offers some benefit in the case of $G = 0$ where no additional DCs can be added, but is actually disadvantageous for $G > 0$ since the location of the initial DCs is not optimized.

While the shown CAGR values are mostly independent of the number of CDNs and applications, the types of services offered which determine the ratio of DC-to-user and DC-to-DC traffic should play an important role. Recall that the overall traffic reduction depends on the tradeoff between DC-to-user traffic reduction and DC-to-DC traffic increment. Consequently, in the next subsection we perform a parametric study evaluating services with different β values.

5.3. Service types: a parametric study

In order to assess the influence of the nature of services on core network congestion and CDN expansion, we ran the multi-period simulation for individual services with different β values assuming a fixed CDN expansion policy. Four applications were considered, each offering a single service with a CAGR of 30% and β values of 0.05, 0.1, 0.125 and 0.5, tuning the DC-to-user and DC-to-DC traffic of that application. Each application was served by 5 CDNs, assuming a greenfield scenario where each CDN was initially comprised of 2 DCs optimally placed to minimize the total core traffic. The CDN growth policy was then fixed to $G=2$ with *TrOpt* DC placement to assess the case with the maximum traffic reduction. The study was performed on the 14-node NSFNet topology for each application individually, for a time period of 15-years. Note only cloud-based traffic, i.e. DC-to-user and DC-to-DC, was considered in this study. The total traffic in the links and the average number of DCs per CDN are shown in Figs. 6(a) and (b), respectively. We can see that for applications with low β values (such long internet video services) which do not require significant inter-DC traffic compared to the DC-to-user traffic generated, adding new DCs was highly beneficial as a mechanism for reducing the traffic in the links by bringing content closer to the users. Specifically, the link traffic CAGR values for the 15-year period for β values 0.05, 0.125 and 0.1 were 18.9%, 22%, 22.6%, respectively. On the other hand, for application traffic with a higher β value, i.e. β=0.5, the traffic growth was optimally supported by expanding current DC locations (i.e. no new DCs were built as no resource benefit was achieved) and the CAGR of the application traffic translated directly to the growth rate of the traffic in the links (30.4%). Note, this is for a greenfield scenario where initial DC locations were chosen for optimal replica placement, and thus no benefit could be achieved by rearranging existing replicas as was the case in Fig. 1(b). The obtained results indicate that CDN growth decisions aimed at reducing core traffic are highly dependent on service type. Namely, building new DC locations can be very effective for low β applications, but such a strategy may not be profitable for CDNs supporting predominantly applications with high β values. Consequently, future CDN expansion strategies should consider the types of applications served by that CDN in order to obtain enhanced and profitable solutions.

6. Conclusion

In this paper, we analyze the effects of content delivery network growth strategies on core traffic evolution. We propose a multi-period optimization algorithm to simulate the core traffic growth for a 15-year period under different CDN growth policies and services types. Our results show that adding new geographically distributed datacenter locations could dramatically reduce the overall link traffic growth rate in the core by bringing content closer to the users. This is true even for the case of random datacenter placement where CDN growth decisions are assumed to be influenced by alternative factors, other than traffic reduction. Specifically, the estimated CAGR over a 15 years period could potentially be reduced by 29% in NSFNet and 15% in EON assuming optimal DC placement and 14% and 10% in NSFNet and EON topologies respectively, considering random placement if a new DC can be added each time the traffic doubles. Consequently, future core traffic growth rates may not be as aggressive as typically forecasted if CDNs continue adding new DC locations. However, the achievable traffic growth rate reduction via CDN expansion is highly dependent on the nature of the services offered, where services with lower inter-DC synchronization requirements can obtain higher benefits than services with higher inter-DC traffic requirements. Results indicate that services with a ratio of 0.5 of DC-to-user to DC-to-DC traffic do not achieve any benefit from adding new DC locations, even if optimally placed, while services with ratios of 0.05 can significantly reduce the CAGR values through CDN expansion. Thus, the future of core network traffic will depend on the nature of the cloud service-based applications that will predominate in the following years and the growth policies of the CDNs which host these applications.

Acknowledgments

The research leading to these results has received funding from the European Commission for the H2020-ICT-2016-2 METRO–HAUL project (G.A. 761727) and it has been partially funded by the Spanish national project ONOFRE-2(TEC2017-84423-C3-1-P, MINECO/AEI/FEDER, UE).

References

[1] Cisco Global Cloud Index: Forecast and methodology, 2015-2020 White Paper.
[2] Cisco Visual Networking Index: Forecast and methodology, 2016-2021.
[3] Sandvine 2016, Global internet phenomena, 2016.
[4] Q. Jia, R. Xie, T. Huang, J. Liu, Y. Liu, The collaboration for content delivery and network infrastructures: a survey, IEEE Access 5 (Jun. 2017) 18088–18106.
[5] F.-J. Moreno-Muro, R. Rumipamba-Zambrano, P. Pavon-Marino, J. Perello, J.M. Gene, S. Spadaro., A evaluation of core-continuity-constrained ROADMs for flex-grid/MCF optical networks,, IEEE/OSA J. Opt. Commun. Netw. 11 (11) (Nov. 2017) 1041–1050.
[6] V. Lopez, L. Velasco, Elastic Optical Networks, Springer, 2016.
[7] H. Yin, X. Zhang, S. Zhao, Y. Luo, C. Tian, V. Sekar, Tradeoffs between cost and performance for CDN provisioning based on coordinate transformation, IEEE Trans. Multimedia. 11 (19) (Nov. 2017) 2583–2596.
[8] Y. Wu, M. Tornatore, S. Thota, B. Mukherjee, Renewable-energy-aware data center placement in optical cloud network, Proc. OFC. (Mar. 2015).
[9] Y. Bang, J-K.K. Ree, K. Park, K. Lim, CDN interconnection service trial: implementation and analysis, IEEE Commun. Mag. 54 (6) (June 2016) 94–100.
[10] I. Goiri, K. Le, J. Guitart, J. Torres, R. Bianchini, Intelligent placement of datacenters for Internet services, in: Proc. of Int. Conf. Distributed Computing Systems (ICDCS), Jun. 2011, pp. 131–142.
[11] S. Ferdousi, F. Dikbiyik, M.F. Habib, M. Tornatore, B. Mukherjee, "Disaster-aware datacenter placement and dynamic content management in cloud networks", IEEE/OSA J. Opt. Commun. Netw. 7 (7) (Jul. 2015) 681–694.
[12] A. Muhammad, N. Skorin-Kapov, M. Furdek, Manycast, anycast and replica placement (MARP) in optical inter-datacenter networks, IEEE/OSA J. Opt. Commun. Netw. 9 (12) (Dec. 2017) 1161–1171.
[13] J. Perello, K. Walkowiak, M. Klinkowski, S. Spadaro, D. Careglio, Joint content placement and lightpath routing and spectrum assignment in CDNs over elastic optical network scenarios, Comput. Commun. 77 (2016) 72–84.
[14] M. Zhu, D. Li, F. Wang, A. Li, K.K. Ramakrishnan, CCDN: content-centric data center networks, IEEE/ACM Trans. Netw. 24 (6) (Dec. 2016) 3537–3550.
[15] S. Borst, V. Gupta, A. Walid, Distributed caching algorithms for content distribution networks, in: in Proc. of Conf. Computer Communication (INFOCOM), Mar. 2010.
[16] Y. Mansouri, S. Azad, A. Chamkori, Minimizing cost of K-replica in hierarchical data grid environment, in: in Proc. of Int. Conf. Advanced Information Networking and Application (AINA), May. 2014, pp. 1073–1080.
[17] J. Sahoo, M.A. Salahauddin, R. Glitho, H. Elbiaze, W. Ajib, A survey on replica placement algorithms for content delivery networks, IEEE Commun. Surv. Tutor. 19 (2) (Nov. 2017) 1002–1026.
[18] M.A. Salahuddin, J. Sahoo, R. Glitho, H. Elbiaze, W. Ajib, A survey on content placement algorithms for cloud-based content delivery networks, IEEE Access 6 (Sept. 2017) 91–114.
[19] H. Wang, Speeding up multi-CDN content delivery via traffic demand reshaping, IEEE International Conference on Distributed Computing Systems (ICDCS), Jul. 2018.
[20] C. Anastasiades, Persistent caching in information-centric networks, IEEE Conference on Local Computer Networks (LCN), Oct. 2015.

[21] M. Elbaham, A traffic visualization framework for monitoring large-scale inter-datacenter network, *International Conference on Network and Service Management (CNSM)*, Oct. 2016.
[22] P. Casas, When YouTube does not work - analysis of QoE-relevant degradation in Google CDN traffic, IEEE Trans. Netw. Serv. Manag. 11 (4) (Dec. 2014) 441–457.
[23] V.K. Adhikari, Measurement Study of Netflix, Hulu and a Tale of Three CDNs, IEEE/ACM Trans. Netw. 23 (6) (2015) 1984–1997.
[24] P. Pavon-Marino, in: Optimization of Computer Networks: A Hands-On Approach, Wiley, 2016, pp. 74–75.
[25] Last accessed. www.net2plan.com.
[26] P. Pavon-Marino, J.-L. Izquierdo-Zaragoza, Net2Plan: an open source optical network planning tool for bridging the gap between academia and industry, IEEE Net. Mag. 29 (5) (Oct. 2015) 90–96.
[27] Last accessed. www.net2plan.com/jom.

Francisco-Javier Moreno-Muro received his Electronics Engineering degree from the University of Granada (Spain) in 2013. In 2014 he joined in Synermet Weather Solution as IT developer and weather researcher. In 2015 he was working as researcher in MATRAS Group of the Universidad de Jaén. Since 2016, he has been working as researcher and Ph.D. Student in GIRTEL group of the University of Cartagena. His research interests include performance evaluation and optimization of communication networks, especially in the SDN/NFV environment.