Energy-Efficient Video-on-Demand Content Caching and Distribution in Metro Area Networks

Omran Ayoub, Francesco Musumeci, Massimo Tornatore and Achille Pattavina
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy (firstname.lastname@polimi.it)

Abstract

The success of novel multimedia services such as Video-on-Demand (VoD) is leading to a tremendous growth of Internet traffic. Content caching can help mitigate such uncontrolled growth by storing video content closer to the users in core, metro and access network nodes. So far, fixed, and, especially, mobile access networks have evolved independently, leveraging logically (and often also physically) separate infrastructures. This means that mobile users cannot access caches placed in the fixed access network (and vice-versa) even if they are geographically close to them, and energy consumption implications of such undesired effect must be investigated. In this paper we perform an evaluation of energy-efficient VoD content caching and distribution under static and dynamic traffic in converged networks as well as in non-converged networks. We define an Integer Linear Programming optimization problem modeling an energy-efficient placement of caches in core, metro and fixed/mobile access nodes where energy is minimized by powering-on and -off caches located in different segments of the network and by performing an energy-efficient VoD-request routing. To deal with problem complexity, we propose an energy-efficient content caching and VoD-request routing heuristic algorithm, which is also adopted under dynamic traffic scenarios. Our results show how deploying caches in the access and metro network segments can reduce the overall energy consumption of the network. Moreover, results show how the evolution towards a Fixed-Mobile Converged metro/access network, where fixed and mobile users can share caches, can reduce the energy consumed for VoD content delivery.

I. Introduction

Internet traffic, requested by both fixed and mobile users, keeps steadily increasing driven by the growing popularity of bandwidth-hungry services and by the increase in number of Internet

users [1]. Network operators must constantly upgrade their infrastructure by massively deploying innovative broadband access technologies, such as fiber-to-the-home (FTTH), to provide most of the population with high access bit-rates. Specifically, Internet's traffic increase is mainly driven by the adoption of the broadband video-streaming services, such as Video-on-Demand (VoD). Cisco predicts VoD to represent approximately 78% of the global consumer traffic by 2019 and as well 80% of global mobile data traffic by 2020 [1], and this trend is expected to grow further in the coming 5G era. To cope with this growth, network operators are also urged to consider new architectural solutions to efficiently distribute the high amount of video contents over the network.

Today, video contents are stored in centralized data centers located in the core segment of the network, far from the users. At the current pace of multimedia traffic growth and without any upgrade in the network infrastructure, the core network will soon be flooded by multimedia traffic with a consequent risk of congestion and service disruption. Such undesirable scenario can be mitigated by pushing contents closer to users. A promising approach, referred as Network Caching, consists of enhancing nodes at the edge of the network with storage and computing capabilities [2]. Taking advantage of Network Function Virtualization (NFV), caching enables edge network nodes to terminate services locally and to offload traffic of the core network [3]. However, the proliferation of caches must be carefully handled from an energy consumption perspective. Several studies, e.g., Refs. [4][5], show that deploying a system of caches in both fixed and mobile networks closer to the users results in two contrasting effects on the energy-consumption. On one side, the *caching energy consumption*, i.e., the energy needed to power the caches where contents are stored, consistently increases. On the other side, the *transport energy consumption*, needed to move data through the network is reduced, as the content is delivered to the users from a closer location.

In this paper we perform evaluations of energy-efficient VoD content caching and distribution in a hierarchical metro networks under static and dynamic traffic scenarios. We propose an Integer Linear Programming (ILP) model and a heuristic algorithm to power-on and -off caches located in different segments of the network, perform content placement and route VoD content requests such that the overall network energy consumption is minimized. Results show, for the considered case study, that deploying caches in the access and metro network segments is decisive to reduce the overall network energy consumption for VoD content delivery. Moreover, we observe that when a multimedia service can be accessed by both fixed and mobile users, the effectiveness of

content caching in the metro/access network segment is hindered by the fact that, as fixed and mobile metro/access networks have been evolved and deployed independently, a cache placed in the fixed metro/access network cannot be easily reached by a mobile network user (and vice-versa), even if the user is geographically close to the cache. In fact, the streaming flow of, e.g., a mobile user trying to reach a cache placed in the fixed network must traverse several additional nodes and links to reach the interconnection point between the two networks. This undesired effect is also known as *trombone effect* [6]. Recent research [7] has defined new architectures for Fixed-Mobile Converged (FMC) networks, where the fixed and mobile metro/access networks are jointly designed and optimized both from a *functional and structural convergence* perspective, and more generally, FMC is raising interest in the context of 5G networks. Clearly, a FMC architectural solution can help in reducing the aforementioned trombone effect [6].

A. *Related Works*

Several studies have already dealt with optimization problems for energy-efficient placement of caches in the network. Ref. [8] is one of the first works investigating the effect of caching on energy-efficiency, proposing different caching strategies for a distributed content delivery network considering content caching in core and metro-aggregation networks. Ref. [9] exploits caching in core nodes, focusing on the definition of efficient strategies to switch off caches and links. Ref. [10] proposes online and offline algorithms to reduce the overall energy consumption of a content delivery network by powering-on and -off data centers through local and global load balancing. Ref. [11] presented an ILP-based model to determine how much bandwidth and energy resources to provision in each cache while minimizing the total cost of caching, considering a core network. We adopt similar approaches, but focus especially on caching in metro and access nodes, as we consider caches are deployed in all nodes of the different network segments (core, metro and access) and are powered-on and -off to achieve energy benefits. Similar to our work, Ref. [12] evaluates energy savings in a multi-level caching systems assuming popular contents are cached at lower level caches and then at caches of higher levels when caches of lower levels are fully utilized. In our model, caches are powered-on when needed, allowing the possibility to power on half of the caches of a network level, and caches deployed at the same network level do not necessary need to store the exact video contents. A feature which we consider can affect the overall energy consumption of the network. In addition, we consider contents can be moved from caches of one segment to another when ever needed, an aspect which has not

been typically accounted in previous work. Furthermore, we investigate on the impact of FMC in reducing the network energy consumption. In our previous work, Ref. [13], we proposed an Integer Linear Program (ILP) model for the problem of energy-minimized cache and content placement and VoD requests routing under a static scenario. We extend the study in this work by considering a dynamic traffic scenario over a large network instance through implementing novel energy-efficient heuristic algorithm for content placement and online VoD request routing algorithm.

More advanced and disruptive architectures for content caching have also been investigated for energy efficiency, Ref. [14] defines an energy-efficient strategy for cache location and content dissemination, considering a logical tree caching hierarchy towards the users. Ref. [15] evaluated the impact of next generation optical access networks on the energy and bandwidth of content distribution but in a locality-aware peer-to-peer network. Refs. [16]–[18] focus on caching in a Content Centric Networking (CCN) scenario. Refs. [16][18] show how a CCN approach can be energy-efficient, especially when popular contents are delivered. Though we do not focus on CCN, these works are relevant as we gather the assumption that every network node can be equipped with storage capabilities. Ref. [19] investigated the popularity-based cache filling policies with the objective of improving the cache performance but without any considerations on the energy consumption of a cache. Finally, Ref. [17] evaluates the impact in terms of performance of shared caching in FMC networks with respect to non-convergent networks. Our assumptions are similar, but we focus on an energy consumption evaluation.

B. Paper Contribution

In this paper we model the problem of energy-minimized cache and content placement and VoD requests routing in FMC networks in static and dynamic traffic scenarios. The main paper contributions can be listed as follows:

- we propose an ILP-based optimization model to evaluate energy benefits provided by an efficient cache utilization under static traffic scenario.
- To overcome the complexity limitations of the ILP model, we propose an energy-efficient heuristic algorithm to power-on and -off caches, perform content placement, and route VoD requests under dynamic traffic scenario.
- We evaluate the benefits granted by fixed-mobile convergence to reduce energy consumption for the specific case of video streaming. Our results quantify the reduction of the overall

energy consumption achieved by utilizing caches in the access and metro network segments. Results show, for the considered case study, that deploying caches in the access and metro network segments is decisive to reduce the overall network energy consumption for VoD content delivery.

The remainder of this paper is organized as follows. Section II describes the network, traffic and energy models considered in our evaluation. In Sec. III we introduce the ILP model used to minimize the overall VoD content delivery energy consumption. In Sec. IV we introduce a heuristic algorithm for an energy-efficient content placement and request routing. In Sec. V we discuss the numerical results considering both static and dynamic scenarios. Sec. VI concludes the paper.

II. Network, Traffic and Energy models

A. Network Model

We consider a hierarchical network topology spanning over three segments, as depicted in Fig. 1:

- The *core* segment, consisting of *core routers* interconnected in a mesh topology and connected to *data centers* hosting video servers.
- The *metro* segment, consisting of *aggregation switches* interconnected by a ring topology. Each metro segment is connected to the core segment through an *edge router*. We consider the edge routers as part of the metro segment.
- The *access* segment, consisting of *access nodes* connected to multiple users. Access nodes act as source of VoD requests. They can be OLTs or DSLAMs for fiber and copper fixed access networks, or eNodeBs for LTE mobile access networks. Note that some access nodes can be directly connected to edge routers.

All the network devices (routers, switches and access nodes) can be equipped with some storage capacity to perform *caching* of content. Obviously, a cache deployment in all network nodes has a high capital cost but its effect on the transport operational expenditure guarantees a return on investment after 1 or 2 years [20]. Specifically, Solid State Drives (SSDs) technology is used for caching video contents in all network nodes. However, we assume that network nodes closer to the users host storage devices with smaller capacity and footprint with respect to nodes far from the users, since the space for the additional storage equipment is more limited

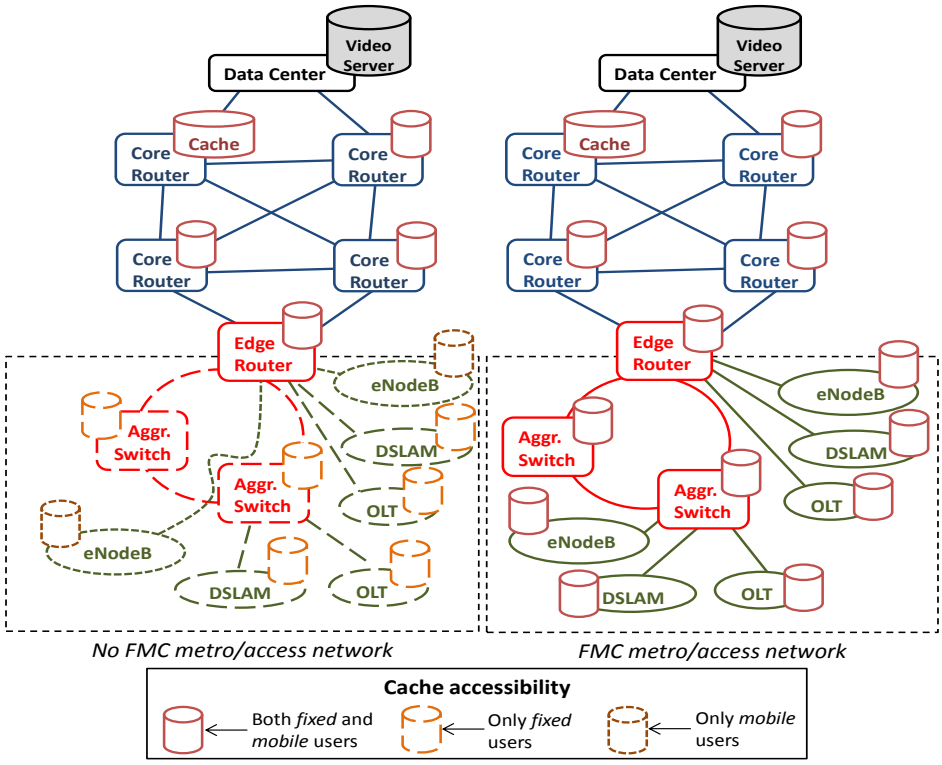


Fig. 1. Examples of *No FMC* network architecture (left) and of *FMC* network architecture (right) spanning over the access (green), metro (red) and core (blue) segments. All the transmission devices can be equipped with caches.

(e.g., cabinets, where DSLAMs or OLTs can be located, are smaller than central offices, where aggregation switches are usually located).

For the metro access segment, we focus on the two different network architectures, as shown in Fig. 1.

1) No FMC metro/access network architecture (No FMC): This network architecture, as depicted in the left-hand side of Fig. 1, models the current mode of operation of the metro and access segments. The fixed and mobile metro/access networks are mostly functionally and structurally independent[1]. In this scenario, we assume that *edge routers* are the first interconnection point between the fixed and mobile networks. Therefore, to avoid trombone effect, caches placed in the fixed (mobile) metro and access network can only be accessed by fixed (mobile) network

[1]Indeed, it can happen that the mobile network uses optical fibers of the fixed network for mobile backhauling and thus a certain degree of structural convergence exists, but the traffic is just tunneled and cannot be accessed, since no functional convergence is provided.

TABLE I

VIDEO RESOLUTIONS AND BIT-RATES FOR THE VIDEO CONTENTS

Video Quality	Video Resolution	Bit-rate R (Mbit/s)
SD	640x460	3
HQ	960x540	4
Mobile	960x640	3
HD	1280x720	6.5

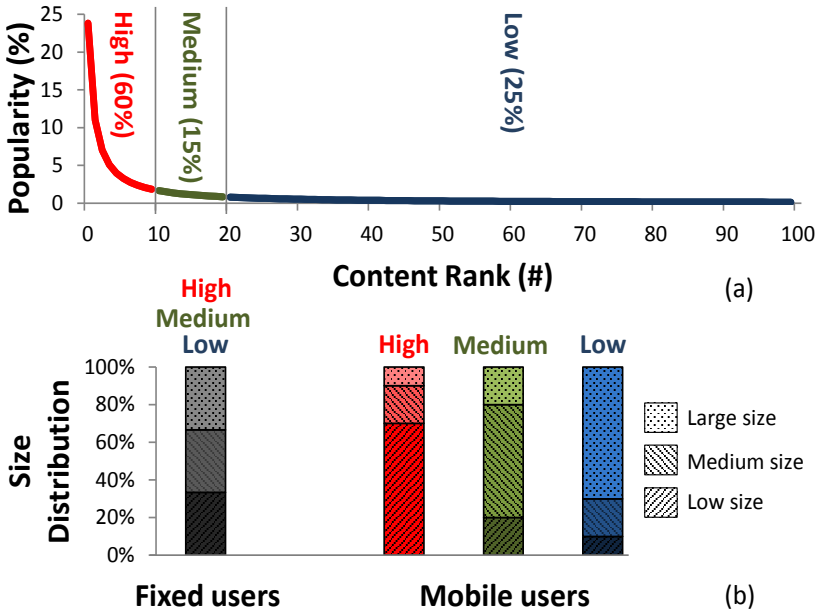


Fig. 2. (a) Content popularity distribution and (b) VoD content size distribution for fixed and mobile users.

users. Instead, caches placed in the edge routers and in the core routers can be accessed by both fixed and mobile users.

2) FMC metro/access network architecture (FMC): This network architecture is depicted in the right-hand side of Fig. 1. In this case, we consider a fully structural and functional FMC network [21]. The metro/access segments are fully shared by the fixed and mobile users. This means that a cache placed in a mobile access node can be accessed by a fixed user by only traversing one or more *aggregation switches*, which are shared among the heterogeneous fixed and mobile networks, and vice-versa. With FMC, we can avoid the trombone effect [6].

B. Traffic Model

Video contents and VoD requests are modeled as follows:

1) Video Content Model: Each content is described by *i)* its popularity, *ii)* its size (byte) and *iii)* its duration. Concerning the VoD content popularity, several studies, such as [11], [22], [23] and [24], show how the popularity of video streaming follows a *Zipf* distribution, where around 80% of content requests are for the 20% most popular contents. This motivates caching popular contents near end-users.

As an example, considering a set M of contents, where $m = 1$ is the most popular content and $m = |M|$ is the least popular content, the probability that the content $1 \leq m \leq |M|$ is requested by a user is defined by the probability density function $h(m) = K/m^{\phi}$, where K is a normalization constant and ϕ is the Zipf distribution parameter. In Fig. 2(a) we show the popularity distribution assumed in this work: considering a total (i.e., content rank) of 100 contents, we assume that the 20 most-popular contents account for an overall 75% of popularity (60% of global popularity corresponds to the 10 most popular contents).

As for the content size, we consider three different categories: low-, medium- and large-size video contents, with file-size of 1 GB, 2GB and 3 GB and with a duration of 20 minutes, 40 minutes and 60 minutes, respectively. We assume that content popularity for fixed and mobile users is different: mobile users require on average more low-size video contents than medium- or large-size ones, while fixed users tend to uniformly choose among the three categories. In Fig. 2(b) we show, for the three categories, the video size distribution for both fixed and mobile user requests. This assumption is motivated by the fact that mobile users are more bandwidth-consumption aware than fixed users, since they usually must deal with very strict weekly or monthly caps on their Internet traffic. Moreover, large-size videos (e.g., long movies) are unlikely to be watched via mobile devices such as smart-phones or tablets. Note that the bit-rate of the request has not affect on the duration of the video content, but only on the video resolution.

2) VoD Request Model: Every VoD request is characterized by *i)* the requested content, *ii)* the bit-rate for the requested content (bit/s) and *iii)* the access node requesting the content. A fixed (respectively, mobile) user requests a specific content according to the fixed (respectively, mobile) popularity content distribution. We assume that a scalable coding technique is used to encode each video content stored and that mobile and fixed users can request the same content, where the contents are encoded 'on-the-fly' and then transmitted with the proper bit-rate to each type of users. One over four possible bit-rates can be associated to a VoD request. Tab. I shows the possible bit-rates and the respective video resolutions [25] [26]. We assume that mobile users always request a content at *mobile* resolution, as usually they access contents through a

TABLE II

ENERGY CONSUMPTION OF NETWORK EQUIPMENT [8]

Transmission Device	Energy Consumption E^{sw} (J/bit)
Core Router	$1.7 \cdot 10^{-8}$
Edge Router	$2.63 \cdot 10^{-8}$
Aggr. Switch	$8.21 \cdot 10^{-9}$
OLT	$1.92 \cdot 10^{-8}$
DSLAM	$1.4 \cdot 10^{-7}$
eNodeB	$2 \cdot 10^{-6}$

mobile device. Conversely, VoD requests of fixed users are assumed uniformly distributed among the four bit-rates shown in Tab. I, since fixed users request the content through heterogeneous devices (e.g., mobile devices connected to Wi-Fi networks, TVs, laptops, etc.).

C. Energy Models

We model the *transport* and the *caching* energy consumption contributions as follows.

- Transport energy consists of *Switching* energy and *Link transmission* energy. The former is due to the switching operations performed by the backplane of switches and routers to forward traffic from/to an incoming/outgoing network interface. We consider this contribution to be proportional with respect to the amount of data processed [27]. In Tab. II we report the switching *energy per bit* E^{sw} of various network equipment[2]. The link transmission contribution is due to network interfaces that are active and transmit data. An active network interface consumes a fixed amount of energy (baseline energy consumption) independently from the actual transmitted traffic, but we assume that all network devices are able to "pack" the traffic in the minimum number of needed interfaces. We assume the usage of Ethernet network interfaces of capacity $C = 1$ Gbit/s consuming $P^{int} = 1$ W.
- Caching energy is composed by a *baseline* contribution, P^b, needed to power-on one cache, and a *storage-dependent* contribution, P^{st}, which is proportional to the amount of data stored in the cache. The values of P^b and P^{st} for a SSD technology storage device of 200 GB are 5 W and $6.25 \cdot 10^{-12}$ (W/bit), respectively [16].

[2]Note that we did not consider a baseline energy consumption of the network equipments, such as the switches and the routers, as we assume these network equipments are powered-on to transport other kinds of traffic.

III. ILP MODEL FOR ENERGY-EFFICIENT CACHING

In this section we present the ILP optimization (ILP) model proposed to solve the problem of energy-efficient caching and distribution of video contents. The associated optimization problem can be stated as follows. **Given** a physical fixed-mobile network topology (as in Fig. 1) and a set of VoD requests, we **decide** the optimal cache and content placement as well as VoD request routing in order to **minimize** the overall network energy consumption. We refer to this ILP-based optimization as *Intelligent Caching* strategy.

1) Sets and parameters: - $\mathcal{G} = (N, A)$ is the graph used to model the physical network topology, where N represents the set of nodes and A the set of bidirectional links.

- The subsets $F, T \subseteq N$ ($T \cup F = N$) represent the set of nodes with forwarding and caching capabilities used to serve VoD requests (we will generically refer to such nodes as *caches*), and the set of terminal nodes, which act as destination for VoD traffic, respectively.

- The subsets $T_{mob}, T_{fix} \subseteq T$ ($T_{mob} \cup T_{fix} = T$, $T_{mob} \cap T_{fix} = \oslash$), are the sets of fixed (OLTs or DSLAMs) and mobile (eNodeBs) terminal nodes, respectively.

- The subsets $F_{fix}, F_{mob}, F_{fixmob} \subseteq F$ ($F_{fix} \cup F_{mob} \cup F_{fixmob} = F$), represent the set of caches that can be accessed by a fixed user, a mobile user, or from any user, respectively.

- The storage capacity of a generic cache $f \in F$ is denoted as S_f. As described in Sec. II P_f^b and P_f^{st} represent the baseline and the load-dependent cache power contributions, respectively. E_f^{sw} is the switching energy contribution.

- M is the set of contents m, each with size B_m and popularity $h(m)$, and $\mathbb{Z}$ is the set of content requests, where each $Z_t^m \in \mathbb{Z}$ represents the number of requests for content $m \in M$ originating in $t \in T$. Every VoD content requested by a terminal node $t \in T$ is associated to a bitrate R_t, chosen among the values in Tab. I.

- C is the capacity of one interface, consuming a fixed amount of power P^{int}.

- δ is a time-interval normalization factor used to evaluate the storage and interface energy consumption given the corresponding power consumption values (P_f^b, P_f^{st} and P^{int}).

2) Decision variables: - x_f (binary) is used to indicate whether the cache $f \in F$ is used ($x_f = 1$) or not ($x_f = 0$).

- l_{ij} (integer) represents the number of active interfaces used in link $(i, j) \in A$.

- $u_{ij}^{m,t}$ (binary) is equal to 1 iff the link $(i, j) \in A$ is used to transmit content $m \in M$ requested by terminal node $t \in T$.

- k_{ij} (integer) represents the amount of data transported on link $(i,j) \in A$. Note that this is an auxiliary variable used to improve the clarity of the ILP model description, and is defined as $k_{ij} = \sum_{\substack{t \in T \\ m \in M}} z_t^m R_t u_{ij}^{m,t}$.
- v_f^m (binary) is equal to 1 iff the cache $f \in F$ is storing content $m \in M$.
- $w_f^{m,t}$ (binary) is equal to 1 iff the cache $f \in F$ satisfies the VoD requests of terminal node $t \in T$ for the content $m \in M$.

3) Objective function:

$$\min \sum_{f \in F} P_f^b \delta x_f + \sum_{\substack{f \in F \\ m \in M}} B_m P_f^{st} \delta v_f^m + \sum_{\substack{f \in F \\ i \in F:(i,f) \in A \\ j \in F:(f,j) \in A}} E_f^{sw}(k_{if} + k_{fj}) + \sum_{(i,j) \in A} P^{int} \delta l_{ij} \quad (1)$$

The first two terms of the objective function account for the *baseline* energy consumption and the *storage* energy consumption of the caches (i.e., the *caching* energy consumption). The third and the fourth terms refer to the *switching* and *link transmission* energy consumption (i.e., to the *transport* energy consumption). The objective of the optimization is to minimize the overall energy consumption. Note that the normalization factor δ is needed to have homogeneous contributions in the objective function. As the switching energy consumption contribution refers to the absolute amount of data switched by each switching device in a given time-interval (and not on the device power consumption), δ can be seen as the average holding time for each VoD request.

4) Constraints:

$$\sum_{m \in M} v_f^m B_m \leq S_f x_f \quad f \in F \quad (2)$$

Eqn. 2 guarantees that the total amount of video content data stored by the cache f does not exceed its capacity and that no data is stored by the cache if the cache is powered-off.

$$w_f^{m,t} \leq v_f^m \leq x_f \quad f \in F, t \in T, m \in M \quad (3)$$

Eqn. 3 assures that a cache f cannot satisfy a VoD request from a terminal node t unless it is powered on and it stores the requested content m.

$$\sum_{f \in F} w_f^{m,t} = 1 \quad t \in T, m \in M \quad (4)$$

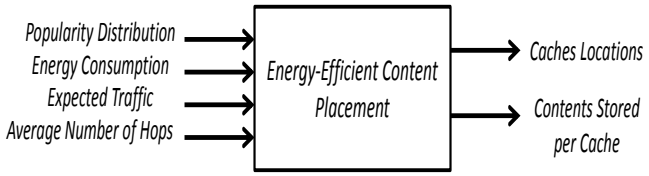


Fig. 3. The input and output parameters of the heuristic energy-efficient content placement approach

Eqn. 4 assures that every VoD request for a content m by a terminal node t is satisfied by exactly one cache f.

$$l_{ij} \geq k_{ij}/C \quad (i,j) \in A \tag{5}$$

Eqn. 5 is used to calculate the number of active network interfaces on link (i,j).

$$\sum_{\substack{i \in F: \\ (i,f) \in A}} k_{if} + \sum_{\substack{t \in T \\ m \in M}} Z_t^m R_t \delta w_f^{m,t} = \sum_{\substack{j \in F: \\ (f,j) \in A}} k_{fj} \quad f \in F \tag{6}$$

$$\sum_{i \in F:(i,t) \in A} k_{it} = \sum_{m \in M} Z_t^m R_t \delta \quad t \in T \tag{7}$$

We set the flow balancing constraints for the caches f (eqn. 6) and the terminal nodes t (eqn. 7). Eqn. 6 refers to the nodes f, that can both generate traffic and forward traffic coming from other nodes. Eqn. 7 refers to the terminal nodes t, that are the destination of video traffic generated by the caches to accommodate the VoD requests.

From a methodological point of view, the fixed-mobile convergence is captured by considering a *unique* (integrated) physical topology, where all caches can be accessed by all terminal nodes. However, two additional sets of constraints are needed when *No FMC* architecture is considered.

$$w_f^{m,t} = 0 \quad f \in F_{fix}, t \in T_{mob} \tag{8}$$

$$w_f^{m,t} = 0 \quad f \in F_{mob}, t \in T_{fix} \tag{9}$$

Eqn. 8 and 9 assure that caches in the fixed network cannot be accessed by mobile terminal nodes and vice-versa.

In addition to ILP-based *Intelligent Caching* strategy, we consider two benchmark strategies: *i) No Caching*, where a centralized video server in the core network stores and delivers all

the contents[3]; *ii) All Caches ON* (*All ON*), where all the caches are assumed as powered-on. Specifically, for the *No Caching* strategy, the decision variable x_f for every $f \in F$ is set to 0 whereas for the *All ON* strategy x_f for every $f \in F$ is set to 1.

IV. Heuristic For An Energy-Minimized Content Placement and VoD Request Routing

To solve larger problem instances, i.e, larger content catalog and/or a larger network, we develop a heuristic for energy-efficient content placement. This approach considers various input parameters and outputs the location of the caches used and the video contents stored in each cache, as shown in Fig. 3, with the objective of minimizing the total energy consumption. Note that we consider a popularity-based content caching model in for the heuristic approach, i.e., contents are sorted in decreasing order of their popularity. The following list summarizes the variables used in this approach:

- V: Total number of contents.
- A, M, C: Number of last content stored in caches located in the access, metro and core segments, respectively.
- $E_{a,i}$, $E_{m,i}$, $E_{c,i}$, $E_{origin,i}$: Energy consumption due to storing and delivering the content of rank (number) i from caches located in the access segment, metro segment, core segment and origin server, respectively.
- $E_a(i,j)$, $E_m(i,j)$, $E_c(i,j)$: Energy consumption due to storing and delivering contents ranked from i to j from caches located in the access segment, metro segment and core segment, respectively.
- U_a, U_m, U_c: Utilization factor of caches located in access, metro and core segments, respectively. The utilization factor of caches of segment x ($x = a, m, c$) is calculated as follows: $U_x = \frac{\textit{Storage utilized in caches of segment x}}{\textit{Total storage capacity of caches in segment x}}$
- $E_{b,a}$, $E_{b,m}$, $E_{b,c}$: The baseline amount of energy required to power ON caches located in the access, metro and core segment, respectively.

The proposed heuristic is divided into two steps. First, an *individual content placement* process is performed which considers each content individually, with the objective of storing

[3]Note that we assume contents are stored in the origin server irrespective of the caching strategy and thus we do not account for its energy consumption

TABLE III

EXAMPLE OF THE OUTPUT OF THE *individual content placement* PROCESS

Segment	Range of contents cached
Access	*from 1 to A*
Metro	*from A+1 to M*
Core	*from M+1 to C*
Origin	*from C+1 to V*

and delivering the content from caches where energy consumption is minimal. Then, a *content placement re-arrangement* process takes place in which contents are reallocated among the caches of different segments to achieve a more energy-efficient content placement.

A. Individual Content Placement

In Fig. 4 we show the flow-chart of the *individual content placement* process and the *content placement re-arrangement* process. First, for every content i, the amount of energy needed to store and deliver the content to all end-users requesting the content from each of the network segments is calculated taking into consideration the total expected number of requests per content (based on the content's popularity). Then, the content placement which guarantees the least energy consumption is chosen. If there is available storage at the caches of the segment chosen, the content is stored in all caches of the considered network segment and the amount of storage available is updated whereas in case no storage is available to store the content, the content is stored at a higher network segment. For clarity, Tab. III shows an example of the output of the *individual content placement* process. After the placement of all contents is decided, the *utilization factors* (U_a, U_m, U_c) of caches of each network segment are calculated, in order to discover if the caches of any network segment are inefficiently utilized. In other words, the *utilization factor* is calculated to avoid powering on caches if only a small portion of their storage capacity is utilized.

B. Content Placement Re-Arrangement

The *content placement re-arrangement* process is needed to balance the utilization of the caches and to power-off the caches that are less-utilized. It is performed according to the *utilization factors* of the caches of the network segments and consists of 3 algorithms, the

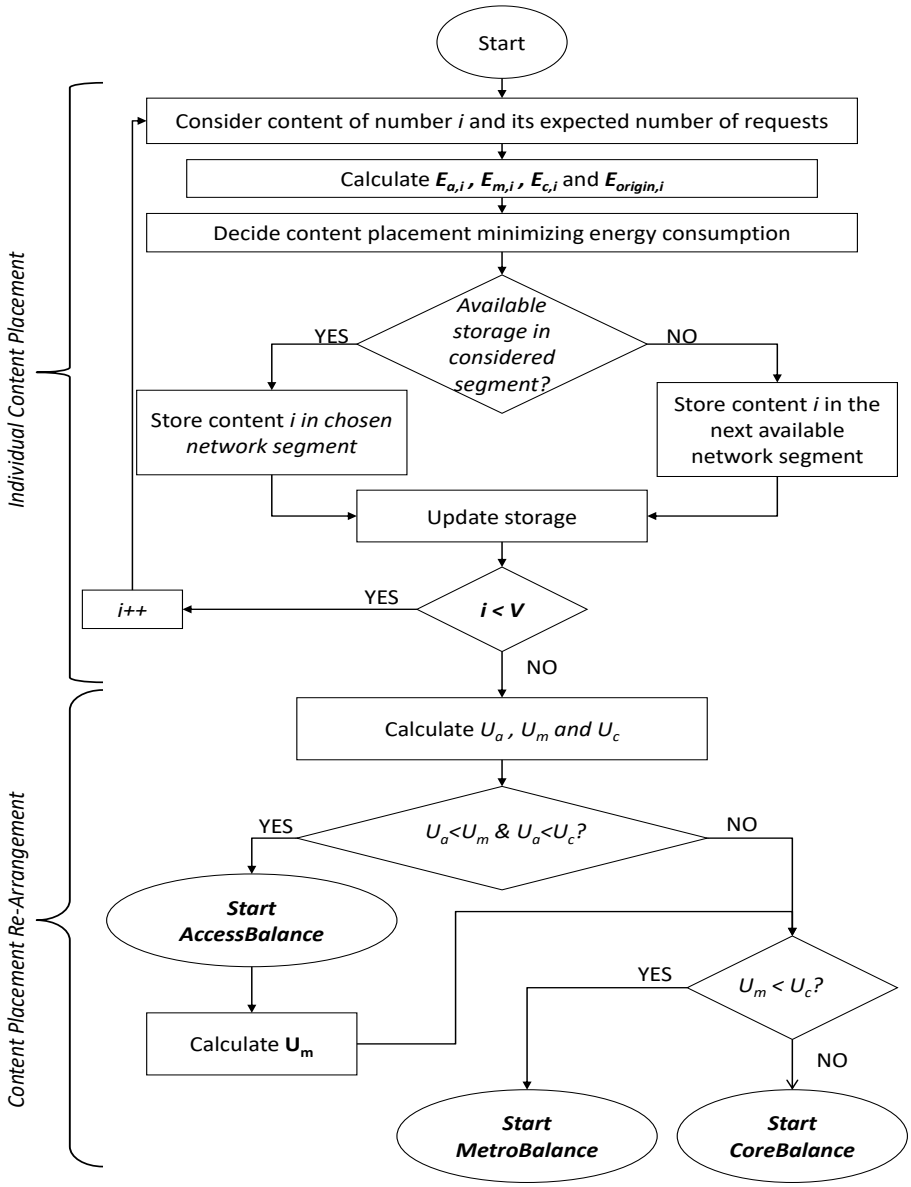


Fig. 4. Flow chart of the content placement heuristic approach

AccessBalance algorithm, *MetroBalance* algorithm and *CoreBalance* algorithm. The 3 algorithms cooperate with the objective of finding a more energy-efficient content placement by balancing the number of caches utilized. If the caches of the access segment have the least utilization factor, i.e., $U_a < U_m$ *and* $U_a < U_c$, the *AccessBalance* algorithm is initiated. Otherwise, if the caches of the metro segment/core segment have the least utilization factor, the *MetroBalance/CoreBalance* are initiated. Note that even after the content placement according to the *AccessBalance* algorithm is performed, the *utilization factors* of the metro and the core caches are calculated for further content placement improvement (as shown in Fig. 4). In this re-arrangement process, the contents which were supposed to be stored in caches of a specific network segment (based on the output

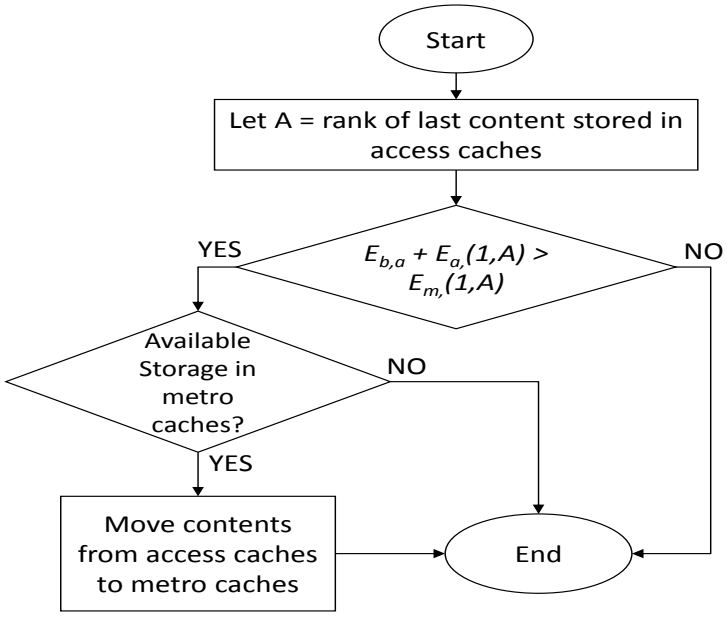


Fig. 5. Flow chart of the $AccessBalance$ algorithm

the *Individual Content Placement* process) can be stored in caches of the lower and/or higher network segment.

1) AccessBalance Algorithm: In Fig. 5 we show the flow-chart of the *AccessBalance* algorithm. First, the total energy consumption of access caches consisting of the baseline energy consumption and the energy consumption of storing and delivering the contents ranked from 1 to A (A being the rank of the last content stored in access caches) is calculated, i.e., *(*$E_{b,a}$ + $E_a(1, A)$*)*. Likewise, the energy consumption of storing and delivering the same contents from the metro caches $E_m(1, A)$ is calculated. In case it is more energy-efficient to store the contents in metro caches and turn off the access caches, i.e., $E_{b,a}$ + $E_a(1, A)$ > $E_m(1, A)$, and enough storage is available in metro caches to store the contents, the contents that were considered to be stored in caches of the access segment are re-allocated and placed in caches located in the metro segment. Otherwise, i.e. if it is not energy-efficient to re-allocated contents, the contents remain stored in their original location.

2) MetroBalance Algorithm: Figure 6 shows the flow-chart of the *MetroBalance* algorithm. First, the total energy consumption of metro caches consisting of the baseline energy consumption and the energy consumption of storing and delivering the contents ranked from $M - A + 1$ to M (M being the number of the last content stored in metro caches and A being the number of the last content stored in access caches), i.e., ($E_{b,m}$ + $E_m(M - A + 1, M)$) is calculated

and compared to the energy consumption of storing and delivering the same contents from the access caches $E_a(M-A+1,M)$. If $E_{b,m}$ + $E_m(M-A+1,M)$ > $E_a(M-A+1,M)$ and enough storage is available in access caches to store $M-A$ contents, the contents are removed from caches located in the metro segment and placed in caches located in the access segment. Otherwise, a loop is initiated to find the number of contents J which guarantees the following conditions:

- Available storage for $M-J$ contents in caches located in the access segment
- Available storage for J contents in caches located in the core segment
- Total energy consumption for storing and delivering contents ranked from $A+1$ to $M-J$ from access caches and contents ranked from $M-J+1$ to M from core caches is less than the total energy consumption of metro caches consisting of the baseline energy consumption and the energy consumption of storing and delivering the contents ranked from $A+1$ to M from metro caches, i.e., if $E_{b,m}$ + $E_m(A+1,M)$ > $E_a(A+1,M-J)$ + $E_c(M-J+1,M)$

If the conditions are met, the contents ranked from $A+1$ to $M-J$ (that were originally stored in caches of the metro segment) are removed and placed in caches located in the access segment and contents ranked from $M-J+1$ to M are stored in caches of the core segment. Otherwise, if the conditions are not met, the contents remain in caches located in the metro caches.

C. CoreBalance Algorithm

A process similar to the *MetroBalance Algorithm* takes place for the *CoreBalance* algorithm, where the contents could be possibly moved to the caches located in the metro or to the origin server to allow turning off caches located in the core segment, given that it represents a more energy-efficient content placement.

D. Energy-Efficient Request Provisioning

In this subsection we present the VoD request provisioning process. Given the content placement performed as described in the previous section, energy-efficient VoD request routing is accomplished as follows.

In Fig. 7 we show the flow-chart of the VoD request provisioning/deprovisioning process. Upon the arrival of a VoD request $r:(m_r,b_r,d_r,D_r)$, from a user D_r at time instant t, the set of caches storing the requested content m_r, represented by S_m, and the origin server are identified[4].

[4]Note that the origin server handles the request of a content in case the content is not cached.

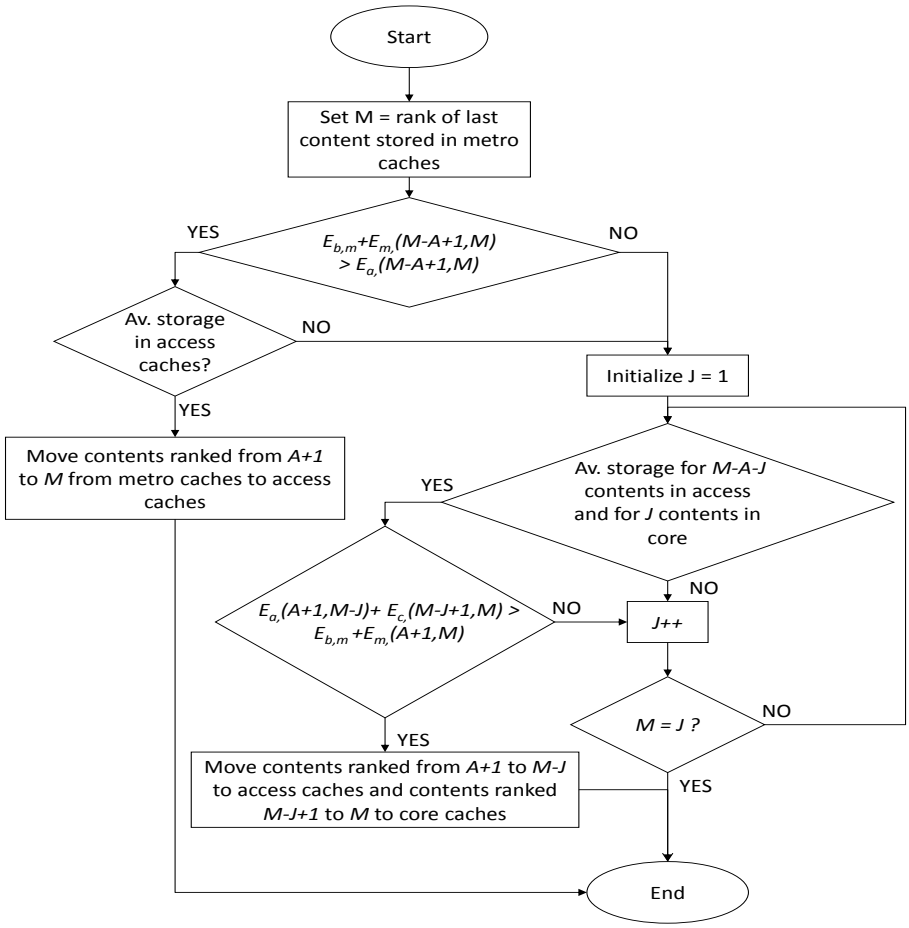


Fig. 6. Flow chart of the $MetroBalance$ algorithm

b_r represents the requested bit-rate whereas D_r represents the destination of the request (which is here the user) whereas d_r is the duration of the content requested. Then, we apply anycast routing and find the possible k-shortest paths having an available bandwidth $B_{av,i}$ greater than b_r towards the nodes where the content is placed. For every shortest path, the amount of energy needed to route the request on the path is calculated taking into consideration if any network interfaces will be powered-on specifically to route the request on the considered path. Then, the paths are sorted in increasing order of energy consumption required to route the request. This motivates packing requests on interfaces that are already powered-on to avoid powering-on new interfaces, even if it means that the request is routed on a longer path. The request r is then provisioned on the path for the duration of the content requested, d_r. Finally, the VoD request is deprovisioned at time $t + d_r$ deallocating bandwidth b_r from path i. If no path is found with enough available bandwidth, the VoD request is blocked.

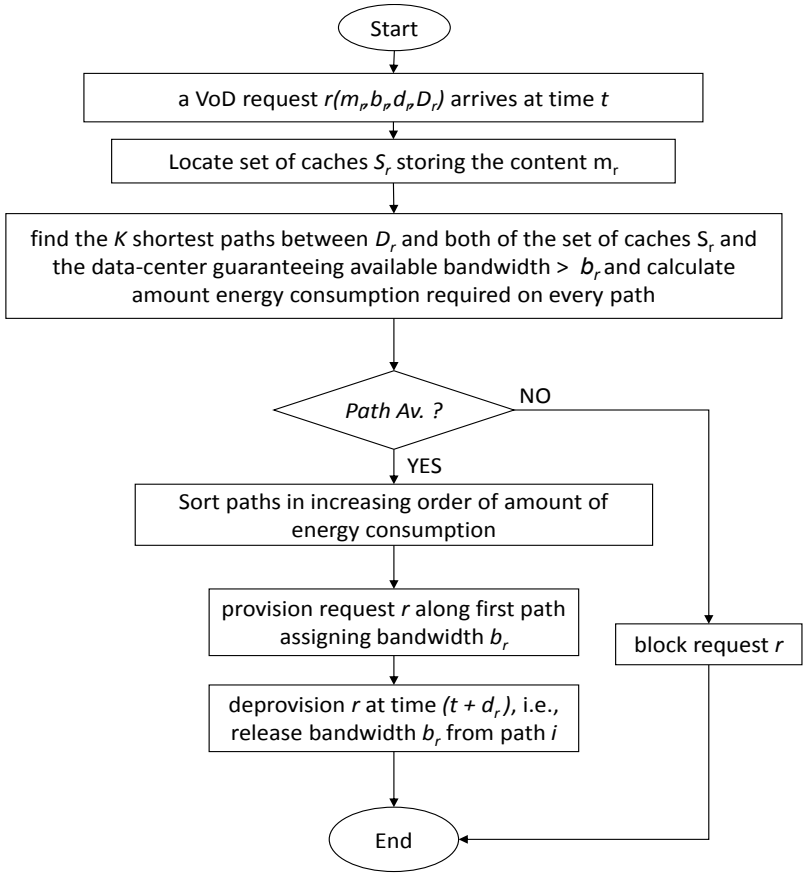
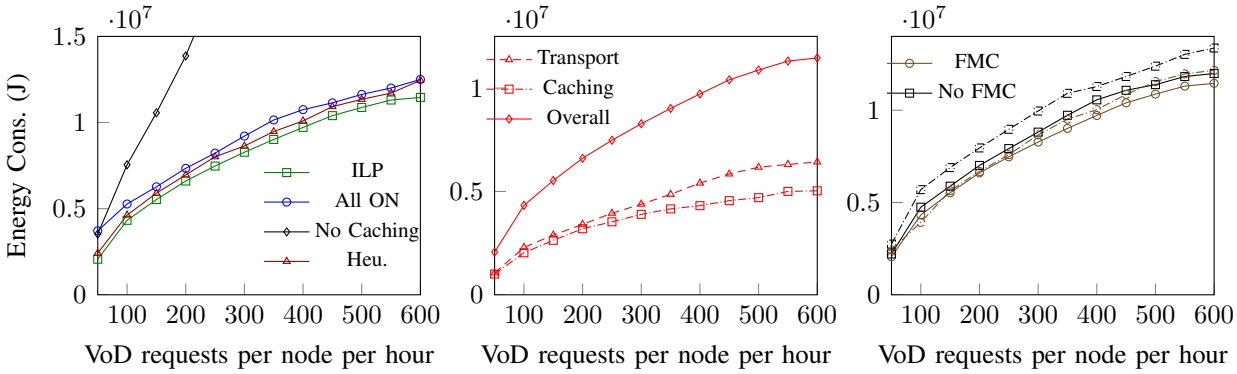


Fig. 7. Flow chart of the provisioning/deprovisioning of a VoD request

V. Numerical Results

A. Case Study

We consider a network topology similar to the one shown in Fig. 1. In the *FMC* architecture both fixed and mobile access nodes can be connected to an aggregation switch and users access caches indistinguishably, whereas in the *No FMC* architecture mobile/fixed users cannot access fixed/mobile caches. The case study varies between the static and the dynamic traffic scenarios. The network and content catalog parameters are shown in Tab. IV. We consider a content catalog whose popularity is Zipf-like distributed ($\alpha = 0.8$). Moreover, content popularity is distributed differently for fixed and mobile users, as shown in Fig. 2. Note that we do not consider a scenario where mobile users can change their locations, however, the approach proposed remains valid in such a scenario by considering the chunk-nature of a VoD request, i.e., the delivery of a video content through different chunks (video segments). In this case, if a mobile user changes location, the remaining chunks are delivered from the nearest cache to the mobile user. Moreover, we



(a) *FMC*: *No caching* vs. *All ON* vs. *ILP* vs. *Heu.* (b) *FMC*: *ILP* breakdown (c) *ILP*: *FMC* vs. *No FMC*

Fig. 8. Energy consumption as a function of the number of VoD requests per terminal node in case of different caching strategies (*No Caching*, *All ON*, *Intelligent Caching*) for the *FMC* architecture (a). (b) shows the caching and transport energy consumption for the *Intelligent caching* and (c) compares the *Intelligent caching* strategy for the *FMC* and *No FMC* network architectures utilizing small caches (dashed lines) and large caches (solid lines).

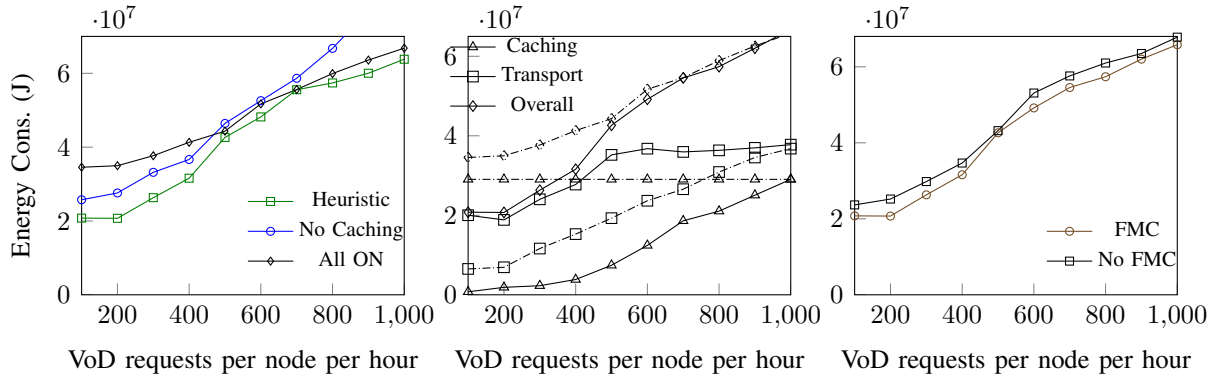
TABLE IV

NETWORK AND CONTENT CATALOG PARAMETERS FOR THE STATIC AND THE DYNAMIC TRAFFIC SCENARIOS

	Static Traffic Scenario	Dynamic Traffic Scenario
Core Routers	4	8
Metro Rings	6	12
Metro Nodes	18	36
Access Nodes	54	108
Number of Contents	100	20000
Total Catalog Size	250 GB	50 TB
Core Cache Size	100 GB	20 TB
Metro Cache Size	50 GB	10 TB
Access Cache Size	20 GB	5 TB

note that for the dynamic traffic scenario, the number of contents and the total catalog size is much greater than that for the static traffic scenario, and accordingly, the size of caches differs in the two scenarios (see Tab. IV). In both instances of the problem, we focus on a time-frame δ of 1 hour. Although moving contents between caches is not in the scope of our work, we note that if contents are to be moved to update the cache and content placement within the caching network, known techniques of CDN providers such as Akamai can be adopted [28].

For the static ILP-based evaluations, numerical results are obtained using ILOG CPLEX 12.4 on a workstation equipped with 8×2.00 GHz processors and with 32 GB of RAM. In all



(a) *Heu.* vs. *All ON* vs. *No caching* (b) *Heu.* (solid) and *All ON* breakdown (dashed) (c) *Heu.*: *FMC* vs. *No FMC*

Fig. 9. Energy consumption as a function of the number of VoD requests per terminal node in case of different caching strategies (*No Caching*, *All ON*, *Heu.*) for the *FMC* architecture (a). (b) shows the caching and transport energy consumption for the *Heu.* and *All ON*. (c) compares the *Heu.* strategy for the *FMC* and *No FMC* network architectures.

cases, optimal results are obtained in computational time in the order of minutes. As previously mentioned, the ILP-based *Intelligent Caching* strategy is compared with the two benchmark strategies: *i) No Caching* and *ii) All Caches ON* (*All ON*).

To evaluate the performance of the heuristic optimization (*Heu.*) approach, we developed a discrete event-driven C++ simulator. Similarly to the static ILP-based case study, the *Heu.* strategy is compared with the two benchmark strategies, *No Caching* and *All Caches ON*. Note that in the static scenario, the requests are known a priori, whereas in the dynamic scenario, the requests are generated dynamically and are Poisson-distributed with an average holding time of 35 minutes for fixed users and 20 minutes for mobile users[5].

B. Discussion

Our evaluation focuses on the energy consumption variation when the number of VoD requests per node per hour increases.

1) Static Scenario: Small Network Instance: We first focus on the ILP evaluations in the static traffic scenario. Figure 8(a) shows the overall energy consumption of the three caching strategies and of the *Heuristic* approach proposed. The *Intelligent caching* strategy always outperforms both the *No caching* and the *All ON* strategies, proving that powering-on only selected caches always

[5]Note that the holding time differs between fixed and mobile users as they follow a different popularity distribution (See Fig. 2).

allows to save energy, as the *ILP* strategy provides an energy reduction ranging between 10% and 45% with respect to *All ON* strategy. Note that the 45% reduction in energy consumption is specifically for low carried load, i.e., at 50 VoD requests per node, as the *All ON* strategy unnecessarily utilizes all the caches of the network, achieving only a small reduction in transport energy as the traffic load is low, but leading for an overall high energy consumption due to powering-on all the caches in all network nodes. Also, note that the *All ON* strategy in general outperforms the *No Caching* one. However, under very low traffic conditions *No Caching* strategy is more energy-efficient than *All ON*, as in the *No caching* scenario the transport energy spent to deliver few contents to a low number of users from the video-server, is lower than the penalty introduced in the *All ON* case to power-on caches in all network nodes. Moreover, we notice that the *Heu.* approach outperforms the *All ON* caching strategy and has a comparable performance with the *Intelligent caching* strategy (5% difference). However, we also notice that both the *ILP* and the *Heu.* strategies only show a slight improvement with the *All ON* strategy for high traffic load, revealing that it becomes decisive to power-on most of the caches to minimize the overall energy consumption of the network.

Figure 8(b) shows the contribution of caching and transport energy consumption on the overall energy consumption when the *ILP* strategy is considered. For a low number of VoD requests per node per hour, the transport and caching energy consumption are almost equal. This is because the *ILP* strategy optimizes caching of contents to limit the increase in transport energy. For a higher number of VoD requests per terminal node, 400 requests and above, the caching energy fluctuates around a constant value while the transport energy continues to increase slightly. This is due to the fact that the *ILP* strategy had already fully-utilized the caches of the network as contents are mostly served by a high number of caches closer to the users (i.e., in the access) leaving no more room for a significant increase in the caching energy consumption yet allowing a small increase in the transport energy consumption as new interfaces are needed to accommodate the additional traffic demands.

We now concentrate on the energy consumption comparison between the *FMC* and *No FMC* architectures (Fig. 8(c)), considering only the *ILP* strategy. Note that in this evaluation, we also consider different dimensions of the caches, to investigate the effect of the cache size on the overall energy consumption. We consider caches of smaller dimension (half the size) at each network segment, with respect to those tabulated in Tab. IV. Generally, the *FMC* architecture is more energy-efficient with respect to the *No FMC* one especially for a higher load, i.e., when it is

more convenient to power-on the caches located in the access nodes, as in the *FMC* architecture they can be accessed by both fixed and mobile users, and thus less number of caches and less replicas of the contents are required, leading for extra energy savings. Moreover, we notice that around 8% of energy reduction is achieved when relatively larger caches are utilized but around 12% in case smaller caches are utilized. This is because the advantage of the *FMC* architecture and the possibility to share caches among mobile and fixed users is more revealed when the storage capacity is limited.

2) Dynamic Scenario: Larger Network Instance: We now focus on the dynamic traffic scenario performed on a larger network instance as shown in Tab. IV. For the content placement, we adopt the energy-efficient content placement shown in Fig. 4 while for the energy-efficient routing the algorithm shown in Fig. 7 is used.

Figure 9(a) shows the overall energy consumption of the dynamic *Heu.* approach and of the *No caching* and the *All ON* strategies. For a lower traffic, i.e. less than 200 VoD requests per node per hour, the *Heu.* approach significantly outperforms the other strategies, demonstrating that even at a relatively low traffic load, caching of contents is decisive to maintain an energy-efficient content distribution. This is because utilizing few caches in the core and metro network does not introduce high caching energy consumption but yet offloads traffic from the origin server to caches located in the core and metro network segments. Moreover, for such traffic load, the *All ON* caching strategy shows the worst performance (highest energy consumption). This is due to the fact that number of caches in a large network is high, and powering ON all caches leads to excessive and unnecessary caching energy consumption. However, we note here that the ratio of powering-on the caches and powering-on the interfaces plays an important role in deciding the degree of penetration of *content caching* in the network. As for higher traffic, the *Heu.* approach and the *All ON* caching strategies show a comparable performance. This is because for high traffic load, the *Heu.* approach utilizes most of the caches of the network, mainly the caches located in the access segment, and thus has a performance similar to that of the *All ON* caching strategy. Moreover, we highlight that the difference in the behavior of the *No Caching* in the Fig. 8(a), which seems exponential, and in 9(a), which increases progressively, is due to the fact that the results of Fig. 8(a) refer to a static scenario of a relatively small case study while the results of Fig. 9(a) refer to a dynamic simulation of a larger case study. To examine more in details the performance of these two strategies, Fig. 9(b) shows the energy contributions of the *Heu* approach and the *All ON* caching strategy. For a low number of VoD

requests per node per hour, the *Heu.* approach's energy consumption is mainly made up of the transport energy consumption, as only caches in higher network levels are utilized. On the contrary, for the *All ON* caching strategy, the overall energy consumption is mainly made up of caching energy consumption. Indeed, the transport energy consumption of the *All ON* caching strategy is much lower than that of the *Heu.* approach. This is because for the All ON caching strategy all caches are fully-utilized, i.e., their storage capacity is 100% utilized, and thus the transport energy is minimized, while for the Heu. approach not necessary all the caches used are fully utilized. This is confirmed as the caching energy consumption of the Heu. approach is less than that of the All ON caching strategy. For a high number of VoD requests per node per hour, the caching energy of the *Heu.* approach increases significantly thus limiting the increase of the transport energy consumption, which fluctuates around a constant value, whereas in the the *All ON* caching strategy, the transport energy consumption slightly increases as all the caches are fully-utilized.

Figure 9(c) shows the overall energy consumption of the *Heu.* approach in the *FMC* and *No FMC* architectures. Indeed, the simulations prove that the *FMC* architecture is a more energy-efficient architecture. As previously discussed, the fact that caches in the *FMC* architecture are accessed by both mobile and fixed users, brings significant energy savings both in terms of caching energy, as less replicas of the same content are required, and in terms of transport energy, as less links are traversed since the content delivery could be performed by any cache.

To examine more in details the behavior of the *Heu.* approach, we show in Fig. 10 its average number of hops and compare it to that of the *All ON* and *No Caching* strategies. Results show that for low traffic load the average number of hops of *Heu.* is comparable to that of *No Caching*, as the *Heu.* utilizes caches of higher network segments. As the traffic load increases, the average number of hops of *Heu.* decreases (i.e., rate of content caching in network segments close to end-users increases), until it becomes comparable to that of *All ON*, signifying that most of the caches are utilized by the *Heu.* approach. Here we also note that a smaller value of the average number of hops has a positive impact on the network latency. For instance, if a request is served from locations closer to end-users, it traverses less hops with respect to the case when it is served from the origin server, and this creates lower variability for latency.

Moreover, since manufacturing the network devices deployed (in our case, caches) in the network requires an energy footprint, i.e. the embodied energy consumption [29], it becomes decisive to account for this contribution if we are to have a fair comparison and to validate

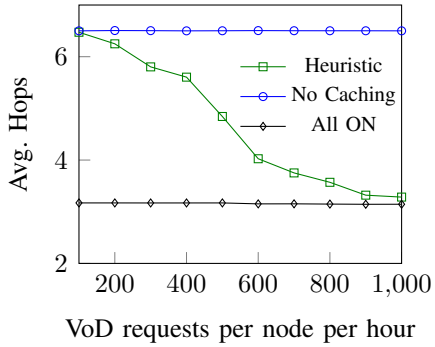
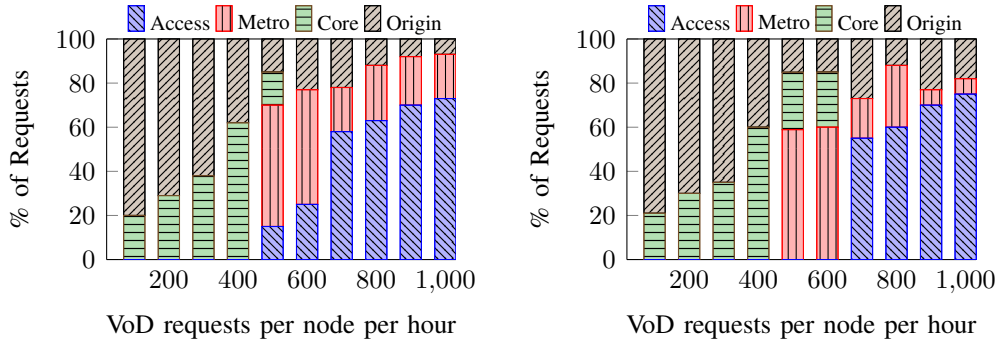


Fig. 10. The average number of hops for *Heu.*, *No Caching* and *All ON* strategies for the *FMC* architecture.



(a) No FMC network architecture

(b) FMC network architecture

Fig. 11. Number of requests served from origin and caches of each segment (core/metro/access) and the average number of hops of all requests as a function of the number of VoD requests per terminal node for the FMC network architecture.

that the approach proposed saves energy with respect to the *No Caching* strategy. We calculated the embodied energy consumption of the caches according to the study in Ref. [30]. The study shows that in 2011, the energy consumption to manufacture a 5 TB storage device was 2926 MJ, and according to Kryder's law [31], which shows that the storage capacity of storage devices doubles every 18 months, the amount of storage which can be manufactured using 2926 MJ is 80 TB in 2018. Accordingly, the embodied energy needed to manufacture 20 TB, 10 TB and 5 TB caches are 731.5, 365.75 and 182.875 MJ respectively. For the case study of the dynamic traffic scenario, the total embodied energy needed to deploy caches at all nodes (as shown in Tab. IV) is 38038 MJ. Referring to the results of Fig. 9(a), we notice that the *Heu.* approach saves, at a medium-traffic load, 10 MJ per hour with respect to the *No Caching strategy*, and thus it will approximately require 160 days to save the energy spent to manufacture all caches deployed in the network.

Figures 11(a) and 11(b) show the percentage of requests served from caches of each network

segment when *Heu.* approach and is performed for the *No FMC* and *FMC* scenarios, respectively. As expected, as the number of VoD requests per node per hour increases, we observe a change of the used caches from the core to the access segment as the hit-ratios of metro and access caches start to increase, as the transport energy contribution becomes more relevant, so its effect is mitigated by pushing the contents closer to the users. However, in the *No FMC* case, this migration first occurs at 400 VoD requests per node per hour whereas in the *FMC* case, the access caches are first used when the number of VoD requests per node per hour is 600. Moreover, we notice that the hit-ratio of the caches in the metro segment in the *FMC* case is lower than that for the *No FMC* case. This demonstrates the energy benefit provided by the opportunity of sharing caches for fixed and mobile users in the *FMC* scenario, as the caches located at the aggregation switches serve both mobile and fixed users and thus allow to power-off caches in the access segment. Note that for a high number of VoD requests per terminal node the caches of the core segment in both architectures are not utilized. This happens because of the nature of the algorithm which tends to re-distribute contents and power off caches with the aim of minimizing energy.

VI. Conclusion

We provided an evaluation of energy-aware cache and content placement strategies that allow to energy-efficiently power-on and -off caches located in core, metro and access network equipment, as well as to efficiently route VoD requests, according to traffic load conditions under static and dynamic traffic scenarios. We developed an ILP optimization model which was adopted in a small network instance. Moreover, to overcome the complexity limitation of the ILP model, we developed a heuristic algorithm for energy-efficient content placement and VoD request routing to be adopted over a larger network instance. We comprehensively evaluated the effectiveness of these strategies on two different network architectures: a *FMC* architecture, where fixed and mobile users can access all the caches deployed in the network, and a non converged network architecture (the current mode of operation, i.e., *No FMC*), where no cache sharing is allowed in the metro and access network segments. We confirmed that, in general, transport energy consumption has higher impact with respect to caching energy consumption as seen in previous works. Indeed, powering-on all the caches distributed in the network allows to save energy with respect to retrieving content from a centralized location, except for very low traffic load conditions. Our results confirm that the proposed strategy, both in static and dynamic traffic

scenarios, allows to save energy in comparison to the cases where all the caches are always powered-on or all the contents are retrieved from a centralized video-server location, as it allows to better manage the trade-off between caching and transport energy consumption. Also, for the first time, we quantitatively showed the impact of *FMC* on network energy-efficiency, in comparison to *No FMC* scenarios, demonstrating that the structural and functional convergence provided by the *FMC* approach is also beneficial for VoD content delivery.

ACKNOWLEDGMENT

The work leading to these results has been supported by the European Community under grant agreement no. 761727 *Metro-Haul* project and the *Lombardy region* through *New Optical Horizon* project funding.

REFERENCES

[1] C. V. Forecast, "Cisco visual networking index: Global mobile data traffic forecast update, 2015–2020 white paper," *Cisco Public Information*, 2016.

[2] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, and W. Snow, "Central office re-architected as a data center," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 96–101, 2016.

[3] M. Tao, W. Yu, W. Tan, and S. Roy, "Communications, caching, and computing for content-centric mobile networks: part 1 [guest editorial]," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 14–15, 2016.

[4] K. Hinton *et al.*, "Power consumption and energy efficiency in the Internet," *IEEE Network*, vol. 25, no. 2, pp. 6–12, Mar. 2011.

[5] C. Jayasundara *et al.*, "Improving energy efficiency of Video on Demand services," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 3, no. 11, pp. 870–880, Nov. 2011.

[6] C. Deliverable, "Assessment of candidate architectures for functional convergence," *V2.0, September*, 2016.

[7] ——, "Final architectural recommendations for fmc networks."

[8] U. Mandal *et al.*, "Energy-efficient content distribution over telecom network infrastructure," in *ICTON*, Jun. 2011.

[9] J. Araujo *et al.*, "Energy efficient content distribution," in *IEEE ICC*, Jun. 2013.

[10] V. Mathew, R. K. Sitaraman, and P. Shenoy, "Energy-aware load balancing in content delivery networks," in *IEEE INFOCOM, 2012 Proceedings*, pp. 954–962.

[11] S. Hasan, S. Gorinsky, C. Dovrolis, and R. K. Sitaraman, "Trade-offs in optimizing the cache deployments of cdns," in *IEEE INFOCOM, 2014 Proceedings*, pp. 460–468.

[12] C. A. Chan, E. Wong, A. Nirmalathas, A. F. Gygax, and C. Leckie, "Energy efficiency of on-demand video caching systems and user behavior," *Optics express*, vol. 19, no. 26, pp. B260–B269, 2011.

[13] M. Savi, O. Ayoub, F. Musumeci, Z. Li, G. Verticale, and M. Tornatore, "Energy-efficient caching for video-on-demand in fixed-mobile convergent networks," in *Green Communications (OnlineGreenComm), 2015 IEEE Online Conference on*. IEEE, 2015, pp. 17–22.

[14] S. Imai *et al.*, "Energy efficient content locations for in-network caching," in *IEEE APCC*, Oct. 2012.

[15] E. Di Pascale, D. B. Payne, and M. Ruffini, "Bandwidth and energy savings of locality-aware p2p content distribution in next-generation pons," in *16th International Conference on Optical Network Design and Modeling (ONDM), 2012*, pp. 1–6.

[16] N. Choi *et al.*, "In-network caching effect on optimal energy consumption in content-centric networking," in *IEEE ICC*, Jun. 2012.

[17] Z. Li *et al.*, "ICN based shared caching in future converged fixed and mobile network," in *IEEE HPSR*, Jul. 2015.

[18] K. Guan *et al.*, "On the energy efficiency of content delivery architectures," in *IEEE ICC Communications Workshops*, Jun. 2011.

[19] S. Dernbach, N. Taft, J. Kurose, U. Weinsberg, C. Diot, and A. Ashkan, "Cache content-selection policies for streaming video services," in *IEEE INFOCOM, 2016*, pp. 1–9.

[20] O. Ayoub, F. Musumeci, M. Tornatore, and A. Pattavina, "Techno-economic evaluation of cdn deployments in metropolitan area networks," in *International Conference on Networking and Network Applications*, 2017.

[21] S. Gosselin *et al.*, "Fixed and mobile convergence: Needs and solutions," in *European Wireless Conference*, May 2014.

[22] H. Li, H. Wang, J. Liu, and K. Xu, "Video requests from online social networks: Characterization, analysis and generation," in *IEEE INFOCOM Proceedings*, 2013.

[23] D. Kim, Y.-B. Ko, and S.-H. Lim, "Comprehensive analysis of caching performance under probabilistic traffic patterns for content centric networking," *China Communications*, vol. 13, no. 3, pp. 127–136, 2016.

[24] P. Seeling, F. H. Fitzek, and M. Reisslein, "Video traces for network performance evaluation: a comprehensive overview and guide on video traces and their utilization in networking research." Springer Science & Business Media, 2007.

[25] netflix.com.

[26] support.google.com/youtube/answer/2853702?hl=en.

[27] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," in *International Conference on Research in Networking*. Springer, 2009, pp. 795–808.

[28] E. Nygren, R. K. Sitaraman, and J. Sun, "The akamai network: a platform for high-performance internet applications," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 3, pp. 2–19, 2010.

[29] X. Dong, A. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Energy-efficient core networks," in *Optical Network Design and Modeling (ONDM), 2012 16th International Conference on*. IEEE, 2012, pp. 1–9.

[30] M. R. Butt, O. Delgado, and M. Coates, "An energy-efficiency assessment of content centric networking (ccn)," in *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*. IEEE, 2012, pp. 1–4.

[31] C. Walter, "Kryder's law," *Scientific American*, vol. 293, no. 2, pp. 32–33, 2005.