A Cloud-based Content Replication Framework Over Multi-Domain Environments

Kostas Katsalis
University of Thessaly, Greece.
kkatsalis@uth.gr

Vasilis Sourlas
University of Thessaly, Greece.
vsourlas@uth.gr

Thanasis Korakis
University of Thessaly, Greece.
korakis@uth.gr

Leandros Tassiulas
University of Thessaly, Greece.
leandros@uth.gr

***Abstract*—Cloud service provisioning on top of virtual infrastructures is of major importance in modern ICT, since it is directly correlated to the way business models are designed and revenue is generated from the cloud service providers. In this work we examine an end-to-end content replication problem over cloud-based multi-technology infrastructures. We extend the classical model where every network node is a potential replica carrier and the link weights represent hops/delay and we examine replication schemes for content that a) is requested by customers belonging in different virtual networks and b) depending on the requester there is different impact on the system operational cost. We examine both centralized and distributed content replication management policies and we evaluate their performance through extended simulations, by means of total cost, the number of object replacements and the number of iterations required.**

Keywords*—Wireless Network Virtualization, Cloud Computing, Content Replication.

I. Introduction

While Software Defined Networks (SDN) design will be in a constant state of mutation the following years, a multifaceted impact in the way that mobile virtual networks are actually build and operate and the way cloud services are provided is expected. We focus our research in end-to-end virtual environments with the following interconnected segments: virtual wireless access networks; virtual optical networks that provide the connectivity required between the wireless domain and the IT resources; and the virtualized back-end datacenter infrastructure. We examine how converged virtual infrastructures, can be used to offer cloud based replication services (CDN-like) and how known replication policies can be exploited by both the physical infrastructure provider and the virtual network operators. Recent works, e.g. [1] and [2], show that a closer collaboration between CDN providers and ISPs will have a proliferative positive effect on the systems operation and end-user performance. Both can jointly take advantage of the already deployed distributed multi-domain infrastructures and also benefit from the advancements in virtualization technology.

In the model that we consider, every domain of the end-to-end path (wireless, optical, datacenter) is virtualized by means of resource virtualization/isolation and in addition is able to host replication facilities, enabling this way replication actions. The replication facilities are accessed by users which belong in virtual networks that use different virtual communication paths with different capacity/cost/network characteristics. We nurture the concept of different costs per object request per virtual operators and per domain. Our objective is to minimize the end-to-end operation cost of the system by exploiting the caching capabilities of the intermediate domains between the users and the datacenter. Our contributions are the following: we formulate an end-to-end replication service provisioning problem over multi-technology virtual infrastructures. We depart form the classical model where every node is a potential replica carrier and the link weights represent hops/delay and we examine replication schemes where the virtual network membership of the requester has impact on the operational cost of the system. We examine both centralized and distributed replication management policies and we evaluate their performance based on extensive simulations.

The rest of the paper is organized as follows: in Section II we present the motivation for this work, related work and the system model, whereas in Section III we formally state the problem under consideration and present the replication management policies used. In Section IV we evaluate through simulations the proposed policies, while we conclude the paper in Section V.

II. Motivation, Related Work & System Model

In an end-to-end virtual environment different virtual networks share the physical resources of different technology domains [3]-[4]. A recent attempt to face the technical limitations of building a framework that will allow multi-technology virtual infrastructures is made by the CONTENT project [5]. The CONTENT project aims to deliver a hybrid solution based on WiFi and LTE access networks and a backhaul WDM metro network that spans until the virtualized datacenter, while the concept of physical resources virtualization is adopted across all the domains to support the IaaS (Infrastructure as a Service) paradigm. In optical networks resource virtualization and slicing can be achieved with WDM [6], or TDMA based Optical sub-lambda [7] techniques. In the wireless domain, the problem of 802.11 access points virtualization is examined in [8] and LTE/WiMax Base Station virtualization work is presented in [9]. With multi-domain virtualization [3], the successful Mobile Virtual Network Operator (MVNO) model can be extended to a Mobile-Optical Virtual Network Operator (MOVNO) model, where virtual communication paths span both the wireless and optical domains to the IT resources.

In this paper we adopt the MOVNO model, in a setting where we assume that the platform is able to build services and offer content replication functionality. As shown in Figure 1, all the different domains are cache-enabled and are able

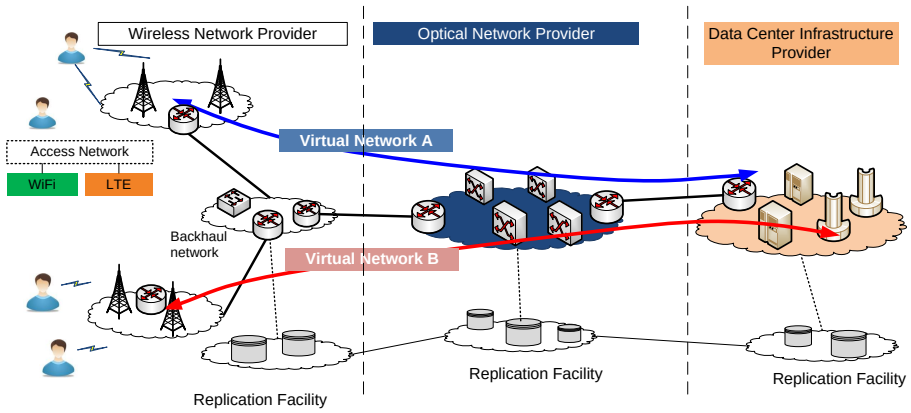

Fig. 1: System model

to perform replication actions. In the MOVNO model that we consider, there is one physical infrastructure provider (PIP) per segment and a single content replication provider (CDN-like), while the physical resources are shared between a number of virtual networks that span end-to-end. The main motivation and questioning behind this work is the following: if all the various segments/domains of an end-to-end architecture are able to perform replication actions and all the mobile end-users request objects from a common pool, but users belong to different virtual networks (VNets) that have different SLAs with the physical infrastructure provider, which is the best replication policy so that the PIP operation cost is minimized?

Content Delivery Networks (CDNs) have been the prevalent method for the efficient delivery of content across the Internet. In order to meet the growing demand for content, CDN providers deploy massively distributed storage infrastructures that host content copies of contracting content providers and maintain business relationships with ISPs. Surrogate servers are strategically placed and connected to ISP network edges [10] so that content can be closer to clients; in [11], the authors highlight that CDN providers and ISPs can indirectly influence each other by performing server selection and traffic engineering operations respectively; and in [12], the authors propose a framework to support joint decisions between a CDN and an ISP with respect to the server selection process. A full-blown ISP-supported CDN service has been proposed in [1] and [13], whereby content is stored and served from within ISP domains.

Current content delivery services operated by large CDN providers (e.g. Akamai and Limelight) can exert enormous strain on ISP networks. This is mainly attributed to the fact that CDN providers control both the placement of content in surrogate servers spanning different geographic locations, as well as the decision on where to serve client requests from (i.e. server selection). These decisions are currently taken without knowledge of the precise network topology and state. Content replication across different network locations and an autonomic cache management framework for future Internet was presented in [14]-[15]. In this work, among others, we also extend the model and algorithms developed in [15] over multi-domain networks. Although research CDNs, such as Coral [16], have proposed distributed management approaches [17], commercial CDN providers have been traditionally using centralized models for managing the placement of content in distributed surrogate servers.

A. System Model

We consider a set of virtual networks $\mathcal{V}$ that span an end-to-end architecture of wireless-optical-datacenter domains. Throughout the paper we will use the calligraphic letters to denote sets and the corresponding capitals for cardinality; for example $|\mathcal{V}| = V$. Also, we denote with $\mathcal{M}$ the set of M objects and with s_i the size (in bits) of object i. We assume that all objects are stored in the datacenter and partially in the optical and the wireless domain replication facilities. For simplicity we assume that the whole architecture is owned by one PIP and let k be an index indicating a domain, where $k \in K$ and $K = \{k : k = 1(wireless), k = 2(optical), k = 3(datacenter)\}$. Also, let S^k note the available storage capacity in the k domain, where we assume that the storage capacity of the datacenter (S^3) is sufficient enough to hold all the objects ($S^3 = \sum_{i=1}^{M} s_i$). We also use additional notations noted in Table I. The main assumption of the system model, is that the operational cost for retrieving an object i depends not only on the domain that the object will be found, but also on the virtual network from which the request was made. This assumption is highly reasonable in today's virtualized networks, since the network and infrastructure providers usually sign for different SLAs and contracts with the Virtual Network Providers and Virtual Network Operators. Under the MOVNO concept, potential cloud-based replication (CDN-like) services that enable this cost differentiation, could offer different incentives to the various Virtual Network Providers and Network Operators.

We note as p_i^k the probability that the object i is available in domain k, where $p_i^k \in \{0, 1\}$. We also assume that $p_i^3 = 1$, $\forall i \in \mathcal{M}$, meaning that every object is always available in the datacenter. We also note as r_i^j the total number of requests per second for object i made by the clients belonging to VNet j, $j \in \mathcal{V}$. The r_i^j is an estimation of the actual request pattern based on observed, historical data (within a given time window) and this estimation is used as a forecast for the future behavior of the users belonging at a VNet. Instead of using user specific request rate per object, we use this term for notation simplicity to expresses directly the entire request flow per virtual network. We also assume that objects can be of different size, while we assume zero delays between objects migration

TABLE I: Notations

Symbol	*Meaning*
d_i^j	the operational cost (per bit) for retrieving object i by a user belonging to VNet j.
c_i^{jk}	the operational cost (per bit) for retrieving object i from domain k by a user belonging to VNet j.
s_i	the size of object i (in bits).
r_i^j	the accumulated number of requests/sec for object i by users belonging to VNet j.
p_i^k	the probability that object i is stored in domain k, $k \in \{1, 2, 3\}$.
V	the number of virtual networks.
M	the number of objects.
S^k	the available storage capacity in domain k (in bits), $k \in \{1, 2, 3\}$.

to a different domain. We note that the analysis of the actual replication operation is not technology agnostic and is very difficult to model. In principle, we consider that the necessary virtualized infrastructure, that a CONTENT architecture for example can provide, offers the capability to deploy cloud based replication services.

III. Problem Statement and Proposed Policies

A. Problem Statement

Access requests trigger the transfer of the requested object from a domain hosting the object to the user where the request was generated. A request by a user belonging to VNet j for an object i cached at domain k has an operation cost:

$$d_i^j = s_i \cdot \sum_{k=1}^{K} \left(P_i^k \cdot p_i^k \cdot \sum_{m=1}^{k} c_i^{jm} \right) \tag{1}$$

where

$$P_i^k = \begin{cases} 1 & \text{if } k = 1, \\ \prod_{m=1}^{k-1} (1 - p_i^m) & \text{otherwise.} \end{cases} \tag{2}$$

If we assume a centralized control mechanism where the controller knows all the cached/replica information in all the domains of the architecture, the objective of the controller would be to find the replication configuration that minimizes the total operation cost D:

$$\text{minimize } D = \sum_{i=1}^{M} \sum_{j=1}^{V} r_i^j \cdot d_i^j \tag{3}$$

subject to

$$\sum_{i=1}^{M} p_i^k \cdot s_i \leq S^k, k = 1, 2, 3 \tag{4}$$

where

$$p_i^k = \begin{cases} 1 & \text{if } k = 3, \\ \in \{0, 1\} & \text{otherwise.} \end{cases} \tag{5}$$

The inequality constraints in Eq. 4, state that the objects that can be stored in each domain cannot exceed the size of the available storage that resides in this domain. Eq. 3 describes the total average operation cost based on a given cache assignment for the delivery of the requested objects from the closest domain that are cached, to the users of each virtual network. Our objective in this work is to minimize the total operation cost of the system (Eq. 3) under the constraint of Eq. 4. However, finding the optimal assignment of the objects in the caches that are available in every domain, even for a static environment, is mapped to the Generalized Assignment Problem. This problem even in its simplest form is equivalent to the NP-complete multiple knapsack problem [18]. In the next section we describe two heuristic cache management policies, one off-line centralized and one on-line distributed for the assignment of the objects in the storages of the domains.

B. Cache Management Policies

We describe two heuristic cache management policies for the assignment of the objects in the available storage of each domain. The first policy is off-line and centralized (*greedy*), whereas the second policy is distributed and on-line (*holistic*).

1) The greedy approach: Authors in [18] and [19] developed several placement algorithms that use workload information, such as distance from the storage points and request rates, to make the placement decision. Their main conclusion is that the so called "*greedy*" algorithm that places replicas based upon both a cost metric and request load, performs the best and is very close to the optimal solution.

Here, we briefly present the greedy algorithm. In each round the greedy algorithm chooses one object to replicate in one of the two intermediate domains (wireless and optical). In the first round, it examines each of the M objects and determines if it must be replicated at each domain. In order to take this decision, it computes the cost gain associated with each object i and selects the one that minimizes the relative total cost. The cost gain of object i depends on the request pattern of each VNet j, the relative probability of finding the object in some domain and the placement cost in that domain. In the second round, the algorithm searches for the second object to replicate which, in conjunction with the stored one, yields the highest cost gain. The greedy algorithm iterates until the available storage capacity of the intermediate domains is full. The greedy algorithm, is an iterative off-line centralized algorithm which requires $\sum_{k=1}^{2} S^k$ iterations (assuming equal size of each object $s_i = s = 1, \forall i \in \mathcal{M}$). It gives solutions of high quality, since its median performance is within a factor of 1.1 - 1.5 of the optimal and around a factor of 4 for the maximum cases.

2) The holistic approach: The second algorithm (*holistic*) is a distributed on-line algorithm and we assume that each intermediate domain has a cache manager, which may update the content of its corresponding cache, by fetching new objects from the datacenter and replacing existing ones upon the detection of a change in the request pattern. We call this process object replacement. A special case of the *holistic* algorithm, where each object is of unit size was presented in [15], whereas here we present the general case.

Given an initial storage configuration of the intermediate domains, the managers, independently and asynchronously, update the contents of their storages towards the global objective. At each iteration a given domain in the holistic approach, say $k \in \{1, 2\}$, executes the following steps:

S1: For each object i, stored in domain k, compute the overall cost increase, $l_i = D_i - D \geq 0$, if object i is removed from domain k, leading hence to a new storage configuration. In this case all the requests for object i will be served by another domain.

S2: For each object i not stored in domain k, compute the overall cost decrease $g_i = D - D_i \geq 0$ achieved if object i is inserted at the storage of domain i, leading hence to a new configuration. In this case a certain amount of requests for object i will be served by domain k, as the closest domain.

S3: Consider as candidate for insertion, the object of maximum cost decrease; say $g^* = \max(\boldsymbol{g}) = g_a$.

S4: Consider as candidates for replacement the object of minimum cost increase. Starting from the minimum one (say $l^* = \min(\boldsymbol{l}) = l_b$), and in ascending order consider that many objects that their total size is greater or equal to s_a (say b and c cause the minimum cost increase $s_a \leq s_b + s_c$).

S5: If the cost decrease of storing the new object is greater than the cost increase of removing the selected objects, perform the replacement, (i.e. *replace* b, c with a).

S6: If the replacement leaves some free space (e^k) in the storage of domain k ($e^k = S^k - \sum_{i=1}^{M} p_i^k \cdot s_i$) compute, for each object i not stored in domain k of size $s_i \leq e^k$, the cost decrease g_i. Store the fitting object of maximum cost decrease. Repeat until no other object could be stored, due to insufficient free space.

S7: Repeat steps 1-6 until no further replacements are beneficial for the system.

In the holistic algorithm only one domain at each iteration (wireless or optical) performs object replacements until a stationary point is reached, where no more beneficial replacements are possible.

IV. Performance Evaluation

In this section, we evaluate through simulations the performance of the two cache management algorithms. We consider a scenario of $M = |\mathcal{M}| = 10^3$ different objects, where the request rate for each object at each virtual network is determined by its *popularity*. Here we approximate the popularity of the objects by a Zipf law of exponents Z_{pop}. Literature provides ample evidence that the file popularity in the Internet follows such a distribution [21]. We denote by $\boldsymbol{\vartheta}_i = \{\vartheta_i^j : i \in \mathcal{M}, j \in \mathcal{V}\}$ the popularity of each object i at VNet j.

In particular, we consider nine typical values for Z_{pop} (popularity exponents of the Zipf distribution) ranging from -1 to 1, i.e. $Z_{pop} \in \mathcal{Z} = \{-1, -0.8, -0.6, -0.3, 0, 0.3, 0.6, 0.8, 1\}$). A Zipf distribution of negative exponent (e.g. $Z_{pop} = -1$) means that out of the M objects the most popular object is the M-th, the second most popular is the $(M-1)$-th and so on, with the first object being the least popular. On the other hand, a Zipf distribution of positive exponent (e.g. $Z_{pop} = 1$) means that the first object is the most popular and the M-th is the least popular. A zero value of $Z_{pop} = 0$ corresponds to equally popular objects. Several measurement based studies [21], shown that the web traffic Z_{pop} value is in the range of $0.64 - 0.84$, but there are other types of traffic (e.g. P2P or video) that follow different Zipf popularity patterns. Since in this work we do not assume a specific application we used a broader range of values for the Z_{pop} value to include a broader set of possible applications. Finally, we assume that in each VNet a total of 50 requests per second is generated. Thus, the request rate of each object at each VNet varies from 0 - 50 requests/sec according to its popularity.

Another factor that we also investigate and is part of our simulation modeling is the so called *locality of interest* factor. The reason is that the locality of similar requests has a significant impact on performance (e.g the efficiency of multicast schemes [22]). This factor quantifies the phenomenon where the popularity of each object may differ from area to area. In our experiments, the workload is tuned from a localized request model, i.e. similar requests originating from the same region, up to a uniform model, and we assume that the VNets are partitioned in $|\mathcal{Z}|$ neighborhoods. Within each neighborhood the popularity of each object i is constant. We assume that the size of each neighborhood v follows a Zipf distribution Z_{loc} of exponent λ_v, where $v = 1, \ldots, |Z|$ and the popularity of objects is given by the corresponding popularity exponent Z_{pop}.

In particular, the first partition $v = 1$ consists of $\lfloor \lambda_1 \cdot V \rfloor$ VNets, where the popularity of each object follows a Zipf law of popularity exponent -1. This set of VNets is computed by choosing randomly $\lfloor \lambda_\vartheta \cdot V \rfloor$ VNets, as long as a VNet has not been already assigned to another neighborhood. Note that $Z_{loc} = 0$ means that the objects are of uniform locality and hence the $|\mathcal{Z}|$ neighborhoods are of equal size ($\frac{V}{|\mathcal{Z}|}$ VNets each). The assumption that locality follows a Zipf distribution is inline with existing literature (e.g. [22]). We are looking for each cache management algorithm a) the *Total Cost* at the stationary point and b) The *percentage difference* of the two cache management policies regarding: the total cost, the number of iterations,the number of object replacements. In all cases the percentage under investigation is calculated by the fraction $\frac{\text{metric (greedy)-metric (holistic)}}{\text{metric (greedy)}}$.

The number of iterations is indicative of the difference of the two algorithms regarding their running time. Regarding the holistic algorithm, the number of object replacements is the number of object fetches (from the datacenter) that have to be performed once the algorithm has converged, i.e. how many items have to be replaced in the cache compared to the initial cache assignment, whereas the greedy algorithm always start from an empty cache and has to fetch every item from the datacenter.

We assume that each domain (wireless, optical and datacenter) has the same operational cost for each object i requested by users belonging to the same VNet j ($c_i^{jk} = c_i^j$, $\forall k \in \{1, 2, 3\}$). This cost ranges from $0.1 - 5$ cost units. Definitely, the $M = 10^3$ objects assumed in this work are not representative for the current Internet, where a content catalog consists of billions or trillions of objects, but this number and the used Zipf popularity is sufficient to quantitatively compare the caching algorithms without overburdening the simulator. Finally, without loss of generality we also assume

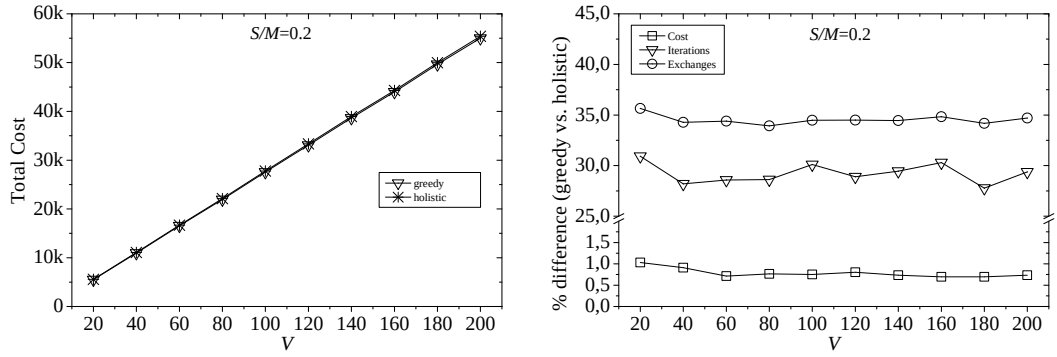

Fig. 2: The performance of the cache management algorithms vs. the number of VNets V.

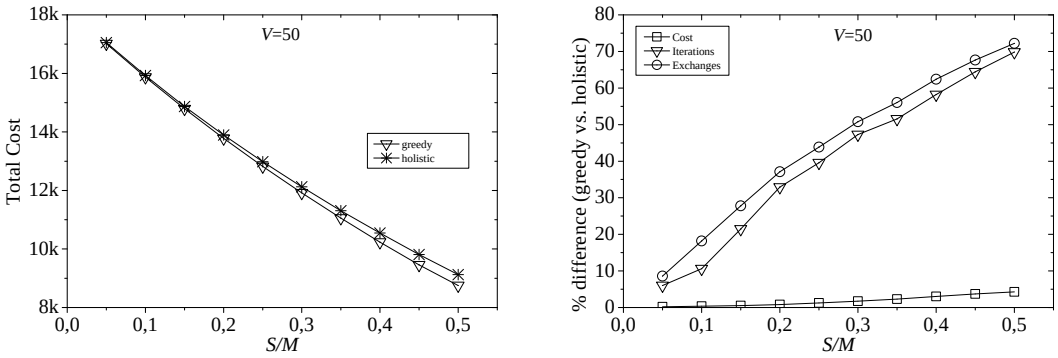

Fig. 3: The performance of the cache management algorithms vs. the fraction ($p = S/M$) of the objects that can be stored in each one of the two intermediate domains (wireless and optical).

that the two intermediate domains (wireless and optical) have the same caching capacity ($S^k = S, \forall k \in \{1, 2\}$), whereas the datacenter has adequate capacity to hold every object. Since the actual performance of the holistic algorithm depends on the initial cache assignment, the depicted values are averages out of 250 executions, where we start from a random initial cache assignment at each instance.

Figure 2 depicts the impact of the number of VNets V in the system. We notice that the Total Cost metric increases linearly with the number of VNets, along with the increase of generated traffic in the system. This means that the two cache management algorithms are not affected by the number of the VNets in the system. Moreover, the distributed holistic algorithm performs less than 1.5% worse than the centralized greedy algorithm, but requires almost 32% less iterations and 35% less object fetches from the datacenter. This implies that the holistic algorithm performs almost identically to the greedy one, but converges faster and produces less overhead cost/traffic (object fetches from the datacenter) from the centralized greedy approach.

Figure 3 depicts the impact of the cache capacity, expressed as the fraction of the objects that can be stored at each one of the two intermediate domains (wireless and optical), on the performance of the two caching algorithms. Regarding the Total Cost metric we observe that the two algorithms perform almost identically, but we observe a linear increase regarding their difference in the number of iterations and object fetches. As we relax the storage capacity constraint and allow more objects to fit in the cache of each domain, the holistic algorithm only needs to make small adjustments in the caches to minimize the cost, whereas the greedy algorithm every time starts with an empty cache and its complexity is strongly coupled with the size of the caches. In more details, in the holistic algorithm the number of object replacements/fetches decreases almost linearly as the capacity of the caches increase, since the availability of more cache slots enables more objects to be stored and hence less replacements are required to reach the selected assignment.

In Figure 4 we investigate the adaptability of the two algorithms as the popularity of the demand patterns change. Particularly, we initially assume that the popularities assigned to the VNets, using a given locality, are given by the vector $\boldsymbol{Z} = (-1, -0.8, -0.6, -0.3, 0, 0.3, \ 0.6, 0.8, 1)$ and at each different experiment this vector changes by a given factor. This factor ranges from 10% to 200%. A change of 10% means that the new vector of popularities is $\boldsymbol{Z} = (-0.9, -0.72, -0.54, 0.27, 0, 0.27, 0.54, 0.72, 0.9)$, whereas a change of 100% transforms the vector of popularities to $\boldsymbol{Z} = (0, 0, 0, 0, 0, 0, 0, 0, 0)$ and a change of 200% inverts the vector $\boldsymbol{Z} = (1, 0.8, 0.6, 0.3, 0, -0.3, -0.6, -0.8, -1)$. As with the previous figures we observe that the holistic algorithm performs almost identical to the greedy, but requires less iterations and object fetches. Note that Figure 4 may also serve as a benchmark for the manager of the system in his decision to reassign or not the cached objects upon the detection of a change in the popularity pattern. Particularly, the difference between the network traffic cost of the initial cache assignment

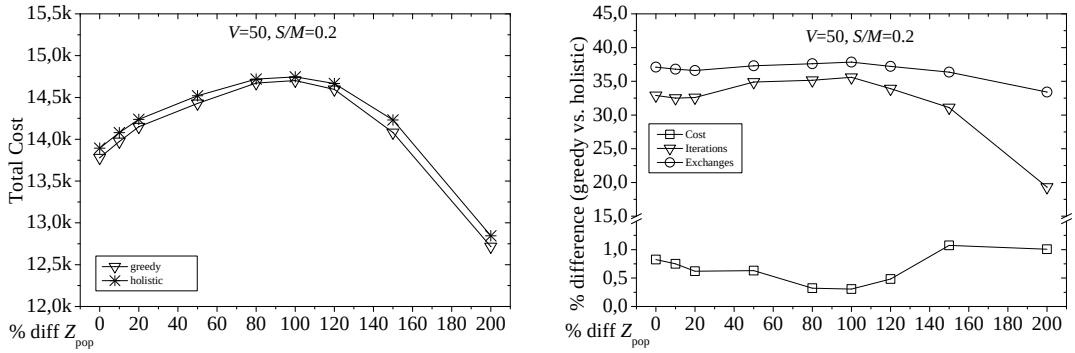

Fig. 4: The performance of the cache management algorithms vs. the popularity exponent Z_{pop}

and the traffic cost after the completion of the algorithms combined with the communication and computational complexity enables the managers to perform or skip the cache reassignment.

V. Conclusions and future work

Cloud services related to content distribution are in the core of today's research, due to the explosion of the mobile usage of Internet and multimedia services. Although it is well known that cloud computing technologies will play a significant role in content delivery, it is less understood how cloud service provisioning will evolve on a global scale in the near future. In this work we get into the insights of content replication strategies and capture the effects of using them in an converged wireless-optical-datacenter virtual environment. Particularly, we compared two different cache management algorithms with regards to their performance, complexity and convergence time. Our numerical results provide evidence that well known distributed approaches give significant performance benefits and reduce the time to convergence when compared to centralized off-line policies. Our imminent future plans is to implement the proposed end-to-end cloud based content replication framework over the facilities (wireless, optical) of the CONTENT project, as well as to investigate new cache management algorithms that will also take into consideration topological constraints of the intermediate domains.

Acknowledgements

This work has been supported by the EU Project n. 318514 "Convergence of wireless Optical Network and IT rEsources iN support of cloud services" (CONTENT).

References

[1] N. Kamiyama, T. Mori, R. Kawahara, S. Harada, and H. Hasegawa, "Isp-operated CDN," in the 28th IEEE INFOCOM,2009, pp. 4954.

[2] B. Frank, I. Poese, Y. Lin, G. Smaragdakis,A. Feldmann, B. Maggs, R. Weber, "Pushing CDN-ISP Collaboration to the Limit". ACM SIGCOMM CCR, 43(3), 2013

[3] M. Chowdhury, F. Samuel, R. Boutaba, PolyViNE: policy-based virtual network embedding across multiple domains, in Proc. of ACM SIGCOMM, workshop on Virtualized infrastructure systems and architectures, 2010.

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. 2008. "OpenFlow: enabling innovation in campus networks". SIGCOMM Comput. Commun. Rev. 38, 2 (March 2008)

[5] Content project: http://content-fp7.eu/

[6] G. Zervas, et al., "Time Shared Optical Network (TSON): A Novel Metro Architecture for Flexible Multi-Granular Services," ECEOC, OSA Technical Digest (CD), Optical Society of America, 2011

[7] M.A Gonzalez-Ortega, Q. Chunming, A. Suarez-Gonzalez, L. Xin, J.-C. Lopez-Ardao , "LOBS-H: An Enhanced OBS with Wavelength Sharable Home Circuits," ICC,IEEE 2010

[8] G. Bhanage, D. Vete, I. Seskar, D. Raychaudhuri, "SplitAP: Leveraging Wireless Network Virtualization for Flexible Sharing of WLANs", IEEE GLOBECOM, 2010

[9] R. Kokku, R. Mahindra, H. Zhang, S. Rangarajan, "Remote Virtualization of a Cellular Basestation", US Patent , US 2012 0002620 A1

[10] A.J. Su, D.R. Choffnes, A. Kuzmanovic, F.E. Bustamante, "Drafting behind akamai (travelocity-based detouring)," in Proceedings of the ACM SIGCOMM, 2006, pp. 435446.

[11] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang. 2009. "Cooperative content distribution and traffic engineering in an ISP network". SIGMETRICS Perform. Eval. Rev. 37, 1 (June 2009)

[12] B. Frank, I. Poese, G. Smaragdakis, S. Uhlig, and A. Feldmann, "Content-aware traffic engineering," in ACM SIGMETRICS/PERFORMANCE 2012 joint international conference on Measurement and Modeling of Computer Systems, 2012, pp. 413414.

[13] K. Cho, H. Jung, M. Lee, D. Ko, T. Kwon, and Y. Choi, "How can an ISP merge with a CDN?" IEEE Communications Magazine, vol. 49, no. 10, pp. 156162, 2011.

[14] V. Sourlas, P. Flegkas, L. Gkatzikis and L. Tassiulas, "Autonomic Cache Management in Information-Centric Networks," in 13th IEEE/IFIP Network Operations and Management Symposium (NOMS 2012), pp. 121-129, Hawaii, USA, 2012.

[15] V. Sourlas, L. Gkatzikis, P. Flegkas and L. Tassiulas, "Distributed Cache Management in Information-Centric Networks", in IEEE Transactions on Network and Service Management (TNSM), vol. 10, pp. 286-299, 2013.

[16] Coral CDN. http://www.coralcdn.org

[17] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, "Donar: decentralized server selection for cloud services," in ACM SIGCOMM 2010, pp. 231-242.

[18] J. Kangasharju, J. Roberts, K. Ross, "Object replication strategies in content distribution networks", Comput. Commun. pp. 376–383, 2002

[19] L. Qiu, V. Padmanabhan, G. Voelker, "On the placement of web server replicas," In Proc. of the IEEE INFOCOM, 2001, pp. 1587–1596.

[20] D. P. Palomar, M. Chiang, "A tutorial on decomposition methods for network utility maximization," IEEE JSAC, pp. 1439–1451, 2006.

[21] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," IEEE INFOCOM, NY, March 1999.

[22] S. Tarkoma, J. Kangasharju, "Optimizing content-based routers: posets and forests," Distributed Computing, vol. 19, Springer, pp. 62–77, 2006.