Advanced Content Caching Schemes and Algorithms

Andreas Papadakis, Theodore Zahariadis, and George Mamais

Abstract—**Current computer networks are challenged by the overwhelming multiplication of available content and the increasing requirements of the content delivery applications. Content caching is already playing its role in content delivery solutions. In this paper we identify the parameters and conditions that affect the performance of content caching solutions and the improvement of the content delivery process. We evaluate caching solutions performing simulation under varying external conditions based on typical service provision considering user requests, content features and network conditions awareness. Based on the simulation results we then describe the mechanisms of a cache federation using various algorithms. We propose novel ways of multimedia data object identification and the exploitation of the network awareness, for further enhancing of the caching capabilities and related service provision.**

***Index Terms*—Content Caching, Content Delivery Network, Content Placement Algorithms**

I. INTRODUCTION

THE exponential increase of exchanged content on the Internet, especially of multimedia nature, is disproportionally higher than the increase of the network and server capacity. The abundance of (mainly multimedia) content is combined with the demand of constantly improved performance in terms of quality, reliability and latency of the networked services and applications, on behalf of the end users. The end users seek for near real time (or real time) access to the content anywhere, anytime using any terminal device.

User requirements such as reduced latency, stable quality of service during content consumption, guaranteed service and data availability set new challenges to the content and service providers and even more to the network providers. One of the solutions employed to enable or facilitate seamless content delivery is the usage of surrogate servers, which are used to cache selected content.

The replication of the content at the network edges, forming the Content Delivery Networks (CDN), results in increased reliability and reduced requirements for centralized processing power and storing capabilities [1]. Furthermore the content access and download times are more predictable and even controllable.

We currently experience a transformation from content replication at the edges of the network (in the form of 3rd party CDNs) to content caching in the internal part of the network with the network providers playing an active role. According to [2], this means that:

1) Content caching mechanism
The content caching mechanism is triggered by requests, in contrast with replication where replicated content is selected a priori. The content can be replaced, based on a subjective and fair content placement algorithm.
2) Network provider role
The network provider is actively involved in caching in order to decrease operational costs. The operator of the caches and the related content placement algorithms are responsible for the management of the cached objects.
3) Type of service
The content provider caching is considered best effort service (in the sense that the cached content may be replaced at any time). In fact the content provider loses strict monitoring and control over the content stored in the network caches.

This enhancement of the network with caching capabilities is performed through the network overlaying principles. Network caching mechanisms are associated with two typical overlays (as identified by [3]): (a) the *Content Delivery overlay* which provides primarily content and secondarily functionality to end hosts in order to improve QoE and/or the infrastructure performance and (b) the Information Routing overlay which is optimizing the content delivery through the network, selecting among multiple possible paths /routes from an original to a destination point, based on cost minimization criteria.

In order to better fit the caching mechanisms into the Future Internet Architecture and achieve reliable performance benefits, we investigate the conditions and parameters that affect their operations. These parameters are related to the content characteristics, the inter-cache collaboration, and the awareness of the network conditions. We initially investigate the influence of specific parameters through experimental simulation and then describe a concrete implementation of a complex cache federation allowing more complex experimentation under realistic conditions.

The structure of the paper is as follows: In Section II, we briefly discuss the background related to caching mechanisms and topologies and the identified affecting factors. In section III we present the experiments we have performed using simulation. Section IV is related to the implementation of

The paper has been submitted in January 2013. This research is partially based on the work performed within the Project REVERIE ICT-287723, which is co-funded by the European Community.

A. Papadakis is with the Department of Electronic Engineering Educators, School of Pedagogical and Technological Education (ASPETE) and with Synelixis Solutions Ltd, Chalkida, Greece (e-mail: apapadakis@ aspete.gr, papadakis@synelixis.com).

T. Zahariadis is with the Department of Electrical Engineering, Technological Education Institute of Chalkida, and with Synelixis Solutions Ltd, Chalkida, Greece (e-mail: zahariad@synelixis.com).

G. Mamais is with Synelixis Solutions Ltd, Chalkida, Greece (e-mail: mamais@synelixis.com).

federated caching scheme and the performance evaluation of typical and advanced content placement algorithms. The last section includes a review and future work.

II. Background

A. Caching Mechanisms

Topologies and caching collaboration mechanisms have been extensively studied [4]. Structured caching topologies impose constraints on the cache topology and (potentially) the content placement mechanisms. The level of imposed structure varies and many topologies have been considered, from the strictly hierarchical to the distributed, including the hybrid solutions. A characteristically distributed caching hierarchy is met in IPTV networks, as described by [5], [6]. Within a distributed caching scenario, DHT (Distributed Hash Table) algorithms and their variations are used to identify the content and the appropriate nodes, without the need of centralized, orchestrating entities.

The question of managing the cache space is related to content (re)placement algorithms and is evaluated based on efficiency and simplicity. The content replacement mechanisms are seeking to replace documents from the cache in order to improve the hit rate. The Least Frequently Used (LFU) and the Least Recently Used (LRU) as well as their variations are the most extensively applied algorithms.

B. Affecting Factors

While each topology and mechanism solution presents its own merits and weaknesses and it is established that the actual benefits increasingly depend on external, dynamically evolving factors. One important factor is related to the supported services which affect the temporal denseness and the statistical properties of the requests.

The characteristics of the content are also pertinent and relate to the typical size of the multimedia files, the number of the corpus of cacheable files as well as their interdependence. For example, in IPTV scenarios cached content is usually large multimedia files, contrary to ordinary web page browsing. The hot spot phenomenon on the other hand, where short – lived content quickly becomes popular and highly requested, creates its own challenges.

The popularity of the content depends also on the homogeneity of the end users. Similar content items are frequently requested within a homogeneous user group, while in a heterogeneous one only dispersed clients are interested in the same content item. One of the most well-known methods to calculate the popularity of a video, which is typically used as the basis for investigations on video server operations, is the Zipf distribution. Following the Zipf distribution, from a set of N multimedia files, stored with a decreasing popularity, the access popularity q_i of the file indexed i is defined as:

$$q_i = \frac{C}{i}, \tag{1}$$

where C is the normalization constant:

$$C = \left(\sum_{i=1}^{N} \frac{1}{i} \right)^{-1}. \tag{2}$$

TABLE I
Factors Affecting Caching Performance

Name	Short Explanation
Services	The supported services have distinct characteristics in terms of request time density, identified patterns which can affect the usage of the caches
Content	The characteristics of the content items, including the number of the cacheable files, their volume as well as their interdependencies can affect the operation of the caches
User group	The homogeneity of the group of the end users affects the distribution of requests among the content items, i.e. the dynamic evolution of content popularity
Network	Awareness of the network characteristics which can permanent as the network infrastructure or dynamic including the bandwidth availability and traffic conditions
Cache mechanisms	The mechanisms (algorithms) employed by the caches for the management and replacement of content can affect their operation
Cache topology	Cache nodes can collaborate with each other forming a federation.

This distribution is completely independent of the number of users and does not really take into account the aging of the individual video [7], [8].

The conditions of the network and the ability of the caching mechanism (the information overlay, as discussed in the introductory section) to be aware of them can play a role primarily in the selection of a single cache node or even the sequence of nodes, in case the content is not initially found. The identified factors are summarized in Table I.

III. Simulation

The design of a network overlay is considered a dynamic process which has to take into account (near) real time conditions and (attempt to) perform the necessary adaptations. Although a set of basic principles are followed, the overlay design is usually considered more of an art than a science. Theoretical models are only currently emerging while experiments can be expensive, complex or even impossible (especially over the production infrastructure).

To mediate risks and to ensure the quality of service, the design process has to be supported and/or validated by extensive simulation scenarios. A simulation framework is needed as well as a set of realistic scenarios.

A. Methodology and Tools

The rationale behind the simulations is to verify the correlation between the performance of the caching mechanism, as perceived through the mean response time, and the (dynamically changing) conditions / factors.

The plethora of selections and parameters that are involved in the design of the network architecture and topology makes the task of creating the simulation scenarios complicated. We have followed a practical methodology, which is used in the multi – parametric problems:

1) We focus on a part of the network designed on realistic assumptions and usage conditions.
2) We identify a set of parameters that affect the operation of the network. Changing these parameters can create a set of network variations. We opt for initially changing each individual parameter in isolation and then changing multiple parameters in parallel.
3) In order to identify the influence of cache deployment on the performance of the network, we consider a set of metrics mainly focused on time delays and network robustness.
4) The simulation scenario is configured and performed for each individual variation resulting in different results.

Network simulation has been an active research field and there is a series of tools for network simulations, both commercial and open source ones. In our experiments we have used the OMNET++ simulation framework, enhanced with the CDN simulator [9].

B. Experiment Preparation

In the basic scenario we consider a typical part of a network of 10 routers, connected in a tree – like fashion of two groups of 5 routers connected with each other through lines of 2 to 34 Mbps bandwidth. The number of clients for this infrastructure ranges from 1000 to 10.000, while the number of the content items ranges from 1.000 to 10.000 files, with their size ranging from 10KB (kilobytes) to 1 GB (gigabyte). The cache capacity ranges from 1% to 100% of the overall volume of the available content, assuming that the product of the number of content files by their size can provide a rough indication of the content volume.

Each individual experiment is configured in detail before its execution. The configuration includes the design of the network topology, the population of the content objects, along with their sizing as well as the as the preparation of the user requests. In order to make reasonable and realistic assumptions, we have studied the characteristics of productive, mainly academic networks.

Network cache deployment is not a black and white choice, as there are multiple interim levels of cache support. This is depending on the number of caches, the cache capacity, and the mechanism for selecting a cache, the cache collaboration mechanism and the content replacement mechanism. We have correlated the number of caches with the network dimensions and specifically the number of the routers, which compose the network (both access and core routers). We have identified three levels for the caching support: 10%, 50% and 90% of the network routers.

In the first case the number of the caches equals 10% of the number of the routers, the network is undercached, then the percentage is set to 50% and then to 90%, the network is overcached (as almost each router is related to a cache). The content replacement mechanism is the LRU (Least Recently Used): when the cache is full, space is made expelling the least recently used content items.

The performance metric is based on the normalized mean response rate (i.e. the time needed to serve content requests). This is the metric used in the following diagrams, although there are supplementary statistics such as the successful and abandoned (content retrieval) cases (described in a qualitative manner).

We begin with a basic network configuration with a set of content items of various sizes which are accessed and consumed by the users of the network with mild intensity.

C. Influence of Caching Level

In this series of experiments we identify the influence of caching into network performance under variable request load. We consider a typical cache collaboration scheme, where the requesting client is redirected to the closest surrogate server in terms of network hops.

If this server accommodates the requested object, it is served to the client; otherwise it is retrieved from peer caches (prioritizing the closest ones). In the case the content is not cached, the content server is contacted. As depicted in Fig. 1, the case with the lowest caching level (10%) starts with the lowest delay which is increasing in parallel with the number of the requests. When the number of requests the scenario of 10% caches inflicts minimal overhead and the caches can effectively support content requested. We observe that the under-cached scenario reaches the maximum value of normalized delay when the number of requests has been increased. On the other hand when the caching capacity is enhanced, the behaviour is reverse: the delay is increased but with lower rate when the number of requests increases. An intermediate behaviour is observed with medium caching capacity. The point of intersection corresponds to the conditions under which different caching options (and subsequent costs) have the same influence on the performance of the network. Increasing caching capacity is meaningful when requests intensity is expected to overcome this level.

D. Influence of Varying File Size

Current applications, such as Video on Demand and IPTV, lead to the increase of the size of the individual content files. In this series of experiments we are verifying the influence of caching, under increasing request load with large content objects. In Fig. 2, the network performance (in terms of normalized mean response time) is depicted with the three caching support levels, when large files are served (mean size of 100MB). It is interesting to observe that the influence of the network caching support follows a similar pattern with the difference that under heavy request load the difference among the performance of the caching levels is increased. At the bottom of the figure, the network performance corresponds to smaller files (up to 10MB).

When the file of size is increased (e.g. file size of 1GB), the influence of caching support is decreased in a disproportional manner; even when the overall volume of the content remains

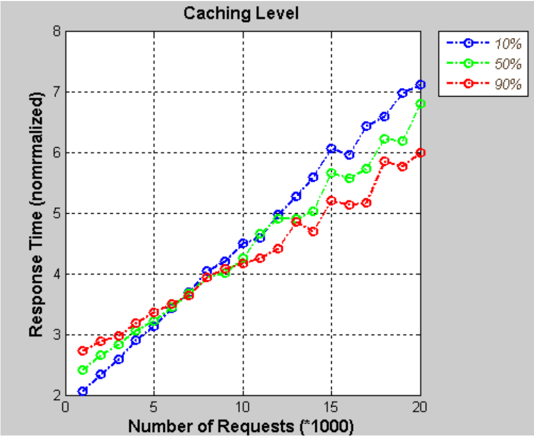

Fig. 1. The performance of the caching scheme is depending on caching level (Mean response time versus Requests). After a short period the highest caching level (905) achieves the best performance.

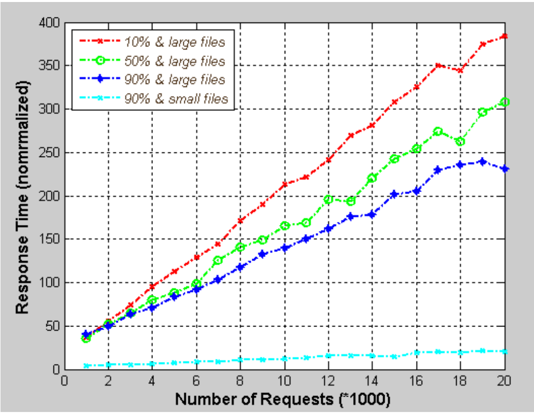

Fig. 2. Caching performance when the size of the files is increased. The performance, in terms of normalized mean response time, deteriorates. The rate of deterioration is higher when the caching level of lower.

the same (this volume can be approached by the sum of the individual file sizes). Not only is the mean response time increased but the aborted attempts are multiplied. This leads to the conclusion that caching performance is more effective in the case of files of small or medium size (up to 100MB). This verifies the current practice of splitting large files into smaller ones (e.g. in the case of MPEG Dynamic Adaptive Streaming over HTTP – DASH [10]). Within the context of caching it makes sense to cache the first part of a large file (e.g. a movie) and when it is accessed by the end users, the cache server will have enough time to retrieve and cache the remaining part.

E. Network Awareness

The performance of content caching and distribution mechanisms is also related to the distance the data travels to reach the final destination. Until recently, it has been difficult for a caching mechanism to be aware in real time (or near real time) of network conditions in the vicinity, the larger region or even wider network segments. All information has been coming from individual (end to end) measurements. Recently, the collaboration between network monitoring and caching entities

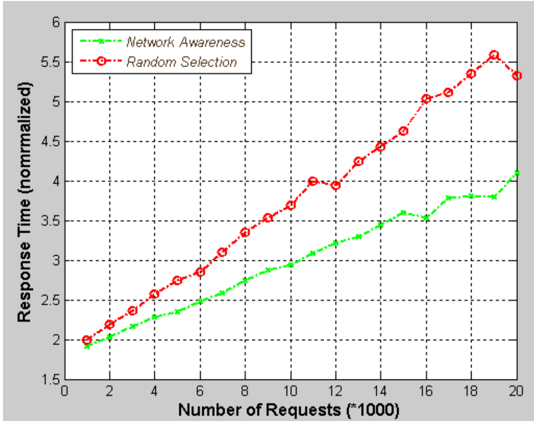

Fig. 3. The performance is dependent on the way the second cache is selected (in the case where the requested content is not present in the first cache). If the cache is selected based on topology awareness the performance is consistently improved than a random selection.

is gaining attention. Deep Packet Inspection (DPI) capabilities and Application Layer Traffic Optimization (ALTO) are two emerging examples, which are expected to play their role in improving caching infrastructures. DPI can provide information on the exchanged content, while ALTO can offer information about the network to applications. This information is relatively stable (e.g. related to the topology of the network or the policies followed by the network operator(s)). It is not associated with data that change in (near) real time, such as the congestion, as this would impose administrative burden; this kind of data can be retrieved through the measurements taken by the applications themselves. The awareness of the network topology supports the selection of the appropriate cache. This selection may be based on the hop count between the client and the cache or the imposed policies of the network operator.

In this experiment (Fig. 3), we have performed two series of simulations: In the first the client is directed to choose the cache which is closer to it (the client node); in the second scenario, the client chooses a cache at random.

In both cases, we may not have a cache hit (with identical probability). In the former case the cache is searching for the nearest cache node which accommodates the requested content, while in the latter case the initial cache node selects another random cache node which can offer the requested content. In the case of random selection, the network performance is aggravated, as depicted in Fig. 3. This aggravation is not constant, but it is consistent. Its value depends on the complexity of the network topology and its size; it is expected that in more complex networks the figure will be larger.

A challenging issue is related to the fact that in a simulation environment the network topology is well known and controllable. This is not true in real application provision scenarios (the cooperation of the network operator is needed but not granted). To this end in our implementation we include a set of indication metrics that can provide useful information on the media delivery overlay (as a subset of the overall network).

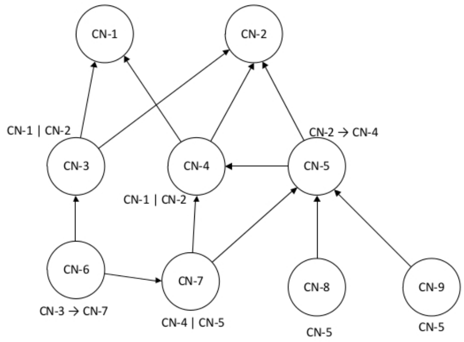

Fig. 4. Mesh topology of cache nodes.

IV. Implementation

Based on the simulation results of Section III, we have implemented an innovative caching scheme, consisting of collaborative Content Nodes, which is described in this Section.

A. Caching Hierarchy

In typical scenarios (such as CDNs) caches collaborate with each other in a well-defined manner. The design of the content delivery overlay topology depends on the number of the overlay nodes and the way they are connected with each other. To enable distributed operation, the Content Nodes (CN) are federated in a directional mesh. The cache federation is depicted in Fig. 4.

The topology of the federation guides the cache behavior when the object requested is not available locally. The cache federation identifies the Content Node which may accommodate the requested object (when it is not found in the first content node – cache).

The content request is forwarded to the content nodes sequentially until the object is found (or the origin server has to be contacted). The sequence is denoted as C1→C2→C3→…CN. In forwarding the request, the initiating cache has the option to enable or disable further forwarding up to an arbitrary number (e.g. like a time-to-live counter which is set at the initial request and which is decremented by one with each additional forwarding).

B. Content Placement Algorithms

The most efficient caching algorithm is the one that discards from the cache the video segment/chunk that will not be needed for the longest time in the future. Since, in principle, it is not possible to predict how far in the future a content segment, chunk or object will be needed, this is generally not implementable. The practical minimum can be calculated only after experimentation, and one can compare the effectiveness of the actually chosen cache algorithm. The algorithms that have been evaluated are:

1) Least Recently Used (LRU). The LRU algorithm discards the least recently used items first. This algorithm requires keeping track of what was used when, which is expensive if one wants to make sure the algorithm always discards the least recently used item. General implementations of this technique require keeping "age bits" for cache-lines and track the "Least Recently Used" cache-line based on age-bits. In such an implementation, every time a cache-line is used, the age of all other cache-lines changes.
2) Most Recently Used (MRU). The MRU algorithm discards the most recently used items first. MRU is most useful in situations where the older an item is, the more likely it is to be accessed. However, taking into account the video popularity metrics and the chunks correlation, MRU does not fit well in our model and it is not further considered.
3) Least-Frequently Used (LFU). The LFU algorithm counts how often an item is needed. Those that are used least often are discarded first. The requests may be weighted with a weight depending on how long in the past the request occurred. The movies with the largest number of requests are stored in the cache.
4) Daily Frequently Used (DFU). The DFU algorithm is a variation of the LFU algorithm that we used setting the length of the LFU to 24 hours. Taking into account the regularity of the daily content requests pattern, the selection of 24 hours as a reference time length is quite reasonable.
5) Peak Hour Frequently Used (PFU). The PFU algorithm is also a variation of the LFU that uses weights for hits that took place during the day. Taking into account the regularity of the daily VoD requests pattern, this algorithm gives higher weights for hits that take place during the peak hours (20:00-22:00), as they have higher probability to be reused in the near future.
6) Randomly Replacement (RR) The RR algorithm is randomly selecting the content object/chunk to be removed from the cache. In this algorithm, any content item is replaced with the same probability.

For the experimental part we have used 10.000 correlated content items of an average size of 1 MB, the hierarchical cache federation of Fig. 4 with 3 layers and the content distribution of Equation 1. For the DFU and the PFU algorithms we have assumed that the requests follow the distribution described in [11], while for the rest we have used the average number of requests.

In Fig. 5 we can observe that the performance depends on the content placement algorithm but the differences are limited. The PFU algorithm is performing better, but it is the most complex one. It is followed by DFU and LFU. The randomly replacement algorithm is not performing so well, but it is the simplest one.

The reason for the relative acceptable performance of the RR is the cache hierarchy. As the same content is already cached in the path towards the server, when a chunk is replace from a child cache and it is needed again, there is a high probability that this chunk still exists in the parent or grandparent cache node. This means that in principle it is needed to traverse all the way up to the origin server.

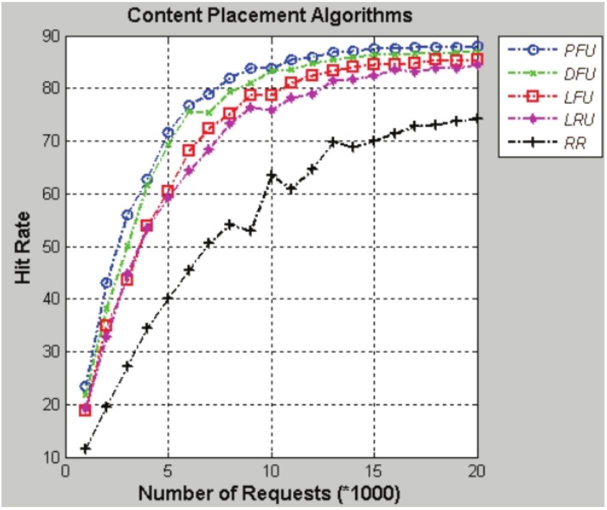

Fig. 5. The performance is dependent on the content placement algorithms. The variations of LFU with shorter windows have the best performance, followed by LRU. Random placement, as expected, lags in terms of performance but leads in terms of implementation simplicity.

C. Content Identification

In a network content caching solution (contrary to content replication), it is highly possible that the same content item which is served by two different servers will be accommodated (twice) by a cache; or that upon a user request for a content object from server A, a cache may not recognize that the same object (coming from server B) is already accommodated and unnecessarily attempt to retrieve it from another cache or the origin server. This is an inherent weakness of current URI (uniform Resource Identifier) - based content identification.

Traditionally files are identified as resources accommodated in a specific server – end point, using a URI. The former part of a URI typically refers to the domain (and port) and the latter specifies a resource accommodated by the server. The domain is associated with an IP (i.e. a server at a specific location of the network), either through a DNS resolution in the simple direct case or through multiple DNS resolutions (involving the CDN authoritative DNS server) in the case of a Content Delivery Network. In a content – oriented scenario (and caching is one of those), the addressability of the content, without (necessarily) the reference to the accommodating server is crucial.

Within the context of enhancing current Internet architecture with caching solutions, we need a solution potentially disentangling the content from the location it (initially) resides (comes from). Solutions assigning unique IDs to available content, independent of the origin server have been proposed, through the usage of self-certified names for content identification [12]. These identifiers belong to a flat namespace and are self-certifying, i.e. they should allow the network or the user to validate that the content fetched indeed corresponds to the content asked. This allows the network to be robust in the presence of malicious nodes and can accordingly relax security requirements and credentials checking when it comes to nodes joining the network or to secure communications among the nodes.

V. Conclusion and Future Work

The increase of the network capabilities cannot satisfy the explosion of the available content, nor the user requests which are exponentially increasing in terms of volume and quality. Content caching can be considered a working solution, which is beneficial both for the content providers and network service providers. The decisions that have to be made are multiple and quite difficult to evaluate beforehand.

In this paper we have explored current caching solutions. In a series of simulations we have explored and offered useful insight of caching solutions towards the increase of content requests. We have identified that the network performance is correlated with the caching level and that the performance of the caching layer is related to the characteristics of the content objects. Furthermore we have proposed and discussed the benefits of innovative content object identification schemes in terms of economizing cache space and reducing administrative burdens.

We have identified that there is an extensive set of algorithms related to content placement, LRU and LFU being the most frequently used. These sets of algorithms can be simple or complicated, light or resource demanding but most of them are in principle based on a common assumption: that the available, cacheable content items (and the subsequent requests) are a priori uncorrelated. In reality this is not always true. A simple scenario, which has already been explored, is related to the splitting of large files in two or more parts: the prefix and the suffix in its simplest case. These content items although independent files are practically correlated with each other. So when an adequate number of requests are made towards the first part, the cache node may trigger the pre-fetching of the second part.

As future work, we are working on coupling advanced content caching scheme with innovative, immersive services. The main challenge in our work is related that the content involved in such services can be static (i.e. created a priori, as currently considered) or dynamic (created in real time).

References

[1] S. Borst, V. Gupta, and A. Walid, “Distributed caching algorithms for content distribution networks,” in *Proc. of IEEE INFOCOM*, 2010.

[2] J. Chuang and M. Sirbu, “Distributed network storage service with quality-of-service guarantees,” *Journal of Network and Computer Applications*, vol. 23, no. 3, pp. 163–185, 2000.

[3] D. Clark, W. Lehr, P. Faratin, S. Bauer, and J. Wroclawski, “The growth of internet overlay networks: Implications for architecture, industry structure and policy,” in *Proc. of Telecommunications Policy Research Conference (TPRC-05)*, 2005.

[4] P. Rodriguez, C. Spanner, and E. Biersack, “Analysis of web caching architectures: Hierarchical and distributed caching,” *IEEE/ACM Trans. Netw.*, vol. 9, no. 4, pp. 404–418, Aug. 2001.

[5] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, and O. Spatscheck, “Network-aware forward caching,” in *Proc. of the 18th international conference on WWW*, 2009.

[6] B. Krogfoss, “Hierarchical cache optimization in IPTV networks,” *IEEE Trans. Broadcast.*, vol. 55, no. 1, 2009.

[7] W. Tang, Y. Fu, L. Cherkasova, and A. Vahdat, “Medisyn: a synthetic streaming media service workload generator,” in *13th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV’03)*, Jun. 2003, pp. 12–21.

[8] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, “Analyzing the video popularity characteristics of large-scale user generated content systems,” *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1357–1370, Oct. 2009.

[9] K. Stamos, G. Pallis, A. Vakali, D. Katsaros, A. Sidiropoulos, and Y. Manolopoulos, “CDNSim: a simulation tool for content distribution networks,” *ACM Transactions on Modeling And Computer Simulation*, vol. 20, no. 2, Apr. 2010.
[10] “Overview of dynamic adaptive streaming over HTTP,” 2011.
[11] T. Qiu, Z. Ge, S. Lee, J. Wang, and Q. Zhao, “Modeling channel popularity dynamics in a large IPTV system,” in *In ACM SIGMETRICS/Performance’09*, Jun. 2009.
[12] T. Zahariadis and E. Quacchio, “Fast content-aware delivery in overlay networks,” *IEEE COMSOC MMTC E-Letter*, vol. 5, no. 1, 2010.

Dr. Andreas Papadakis received his PhD and Dipl. –Ing. degrees in electrical and computer engineering from the National Technical University of Athens, Greece. He is an Assistant Professor in the School of Electronics Engineering Educators (ASPETE) in Athens, Greece. In the past he has worked in the private sector, as a managing consultant and project manager in the private sector. He has also worked as special scientist in the area of communications in the Ministry of Interior, where he has been involved in the Greek Public Sector’s telecom infrastructure. He is interested in Future Internet Architecture, multimedia encoding and perceived quality evaluation. He has significant experience in European RTD projects. He has published more than 30 papers in magazines, journals and conferences and many book chapters. He has been a reviewer in IEEE and ACM journals and magazines. He is a member of the Technical Chamber of Greece.

Dr. Theodore Zahariadis received his Ph.D. degree in electrical and computer engineering from the National Technical University of Athens, Greece, and his Dipl. -Ing. degree in computer engineering from the University of Patras, Greece. He is heavily involved in the EU Future Internet Assembly (FIA) activities; he coordinates the Future Media Internet subgroup as caretaker, while he participles in the Real World Internet/”Internet of Things”, the Future Services Internet and the Secure Future Internet subgroups. He is also the coordinator of the Networked Media Group 1: Media Delivery Platforms cluster, a cluster of more than 15 EC projects. In the past, he has been with Ellemedia Technologies as the Technical Director, Hellenic Aerospace Industry (HAI) as Chief Engineer, Lucent Technologies/Bell-Laboratories, NJ as a Senior Consultant, Intrasoft, Intracom, and NTUA as Senior Researcher. He is also an Assoc. Professor at the Technological Educational Institution of Chalkida. Dr. Zahariadis has published more than 80 papers in magazines, journals and conferences and he is the author of two books and many book chapters. He has been a reviewer and principal guest editor in many IEEE and ACM journals and magazines and he is a Technical Editor of the IEEE Wireless Communications Magazine. He is a member of the IEEE and the Technical Chamber of Greece

Dr. George Mamais received his diploma in Electrical and Computer Engineering and his Ph.D. from the National Technical University of Athens. He brings with him more than 16 years of experience as developer/technical project manager/project manager in numerous projects in the fields of telecommunications, GSM, POTS and defence software. In Synelixis, Dr. Mamais leads research in content centric networking, telecommunications product development and has successfully delivered solutions for GSM operators, billing system vendors and equipment manufacturers.