Design Principles of an Operator-Owned Highly Distributed Content Delivery Network

Stella Spagna, University of Pisa

Marco Liebsch, Roberto Baldessari, Saverio Niccolini, and Stefan Schmid, NEC Laboratories Europe

Rosario Garroppo, University of Pisa

Kazunori Ozawa, NEC Corporation

Jun Awano, NEC Central Research Laboratories

ABSTRACT

Mobile network operators are experiencing a tremendous increase in data traffic due to the growing popularity of bandwidth-intensive video services. This challenge can be faced either by boosting the capacity of the network infrastructure, or by means of offloading traffic from the backhaul and core network and serving contents from distributed cache servers close to the users. Network operators can extend the coverage of traditional CDNs by making usage of caching locations much closer to the users than traditional CDNs. Additionally, network operators can optimize the caching and delivery of contents by exploiting the complete knowledge of their network for designing a cost-effective infrastructure able to achieve both improved user satisfaction and cost savings. This article provides thoroughly justified design principles for a highly distributed operator-owned CDN while focusing on four key aspects: the optimal location of cache servers, mechanisms for request routing, content replica placement, and content outsourcing and retrieval.

INTRODUCTION

Video services in the Internet are achieving tremendous success. Ubiquitous access to such services results in mobile network operators facing capacity problems in their network infrastructure due to the rapid growth in data traffic. While Reelseo forecasts a fourfold increase in over-the-top (OTT) traffic by 2013 and Cisco forecasts a 15-fold increase of mobile video traffic by 2015, it is clear that operator revenues will not be following such a trend and will thus not be sufficient to offset the investment levels necessary to grow bandwidth so fast.

A major capacity problem in mobile carrier networks results from the centralized nature of the network architecture, which requires all traffic from and to user equipment to traverse the backhaul and core network up to the centralized packet data network gateway (PDN GW). Since boosting the transport capacity of the backhaul and core network is very costly, network operators tend to adopt technology for content delivery networks (CDNs) [1, 2] and plan the deployment of distributed cache servers close to users, thereby offloading traffic from the core network. A transport cost reduction and shorter delivery paths to the users represent the desirable effects of this strategy.

How efficiently a network-operator-owned distributed CDN can save resources depends on the specific operator topology, the CDN-specific design choices, as well as the degree of integration. To enable direct access to entry points and retrieve content from distributed cache servers, mobile operators must decentralize their core network, by either distributing PDN GWs or adopting the technology for selected IP traffic offload (SIPTO) using local GWs (L-GWs) [3]. The CDN must provide efficient means to route requests to the most suitable CDN entry point and cache server to bypass congestion. The challenge of content distribution is to balance storage costs and availability of popular content on cache servers near a requesting user.

Based on a thorough analysis of the above cited issues, this article provides the design principles for network-embedded operator CDNs related to cache servers placement, content outsourcing, replica placement, and request routing. These principles are substantiated by simulation and analytical evaluation.

The rest of the article is structured as follows. We provide an overview of the base CDN functionalities and present a short summary of the existing solutions. We summarize the design goals that served as a base for finding the CDN design principles for a mobile network operator

0163-6804/13/$25.00 © 2013 IEEE

described. We present the proposed implementation of an operator-owned CDN, which allows meeting the depicted design principles, and then conclude this article and summarize the lessons learned.

BASELINE FUNCTIONALITIES AND DESIGN CHOICES FOR AN OPERATOR-OWNED CDN

The first important choice for the design of an operator owned CDN is the physical placement of cache servers. They can be placed at locations ranging from the radio access network to the core network with trade-offs impacting the total capital expenditure (CAPEX) and operation expenditure (OPEX).

The remaining design choices are related to specific functionalities of the CDN:

- Content acquisition and ingestion
- Content distribution
- Request routing logic
- CDN management

Content acquisition and ingestion refers to:

- The mechanisms describing how content enters the CDN from the source (e.g., the content provider)
- The preparation of the content for distribution (i.e., transcoding to multiple formats with multiple codec rates)

Choices around such functionality are not addressed in this article as they are mainly imposed by commercial and service-specific considerations.

Content distribution is one of the core functionality of a CDN, comprising:

- Content outsourcing and placement (i.e., the outsourcing strategy of content replicas to cache servers)
- Content retrieval (i.e., the moving of content replicas among different cache servers)
- Content selection and delivery (i.e., the selection and delivery of the right content from the cache server to the end user)

The content outsourcing strategy can either proactively push content replicas to the cache servers or pull on demand. In the first case a replica placement algorithm needs to be designed; in the second case the content is cached only after a user request. The impact of these two different strategies are on the perceived latency of accessing the content from the user perspective vs. the algorithmic complexity of the involved mechanisms.

The content retrieval strategy deals with the question how to retrieve content in case the local cache server has no replica yet. Answers to this question can be either to use a cooperative scheme, such as a peer-to-peer (P2P) multisource retrieval among different cache servers, or a non-cooperative one (i.e., always retrieve the content from the origin server). Implications of this choice are on the possibility to support load balancing, quickly adapt to overload situation due to flash crowds, and so on.

Of notable importance to a CDN design is the request routing logic, that is, the set of metrics, algorithms, and mechanisms that route user requests to an appropriate cache server. The design choice related to such logic spans from the selection of metrics for request routing (e.g., network proximity, client perceived latency, cache server load) to the selection of request routing algorithms.

Last but not least, CDN management is the functionality that allows the CDN service provider to gauge the efficacy of the CDN with respect to its customers. This functionality involves the selection of a bouquet of metrics to evaluate the performance of the CDN. Commonly used metrics are cache hit ratio, origin server bandwidth, latency, cache server utilization, and availability. This article focuses on the key design issues that are most relevant to a network-embedded operator CDN in order to maximize the user satisfaction and minimize the operational costs:

- The physical placement of cache servers
- Content outsourcing
- Content replica placement
- Request routing logic

These design choices are highly correlated with each other.

The content outsourcing strategy can either proactively push content replicas to the cache servers or pull on demand. In the first case a replica placement algorithm needs to be designed; in the second case the content is cached only after a user request.

RELATED WORKS

The goal of this section is to present, for each of the baseline functionalities we investigate in this article, the main approaches present in the literature that are also applied on the CDN market.

Cache server placement: There are different existing cache servers placement strategies. Theoretical approaches such as minimum k-center problem and k-hierarchically well separated trees (k-HST), model the cache servers placement problem as a center placement problem as defined in [4, 5]. These approaches are very complex to implement; therefore, some heuristics [6] have been proposed in order to provide reasonable solutions with lower computation costs.

Content outsourcing: Most popular CDN providers (e.g., Akamai) use a non-cooperative pull-based approach. The drawback of this approach is that an optimal server may not always be chosen to serve content request, as shown in [7] (i.e., the optimal server is temporarily busy, overloaded, or unreachable). Different from the non-cooperative approach, the cooperative pull-based approach allows cache servers to cooperate with each other (i.e., using a P2P communication model). An academic CDN, Coral [8], has implemented the cooperative pull-based approach using a variation of a distributed hash table (DHT).

Content replica placement: Most of the replica placement algorithms are greedy solutions that try to find a trade-off among loads, sizes of the cache servers, and the latency of retrieval.

In [9] the authors have presented a set of greedy approaches where the placement is done by balancing the load and sizes of the cache servers. A self-tuning, parameterless algorithm has been presented in [10]. The price to pay is the increase of the algorithm complexity and response time.

Request routing logic: To differentiate from the most common and simple round-robin algorithm, Cisco Distributed Director has implemented an adaptive request-routing algorithm

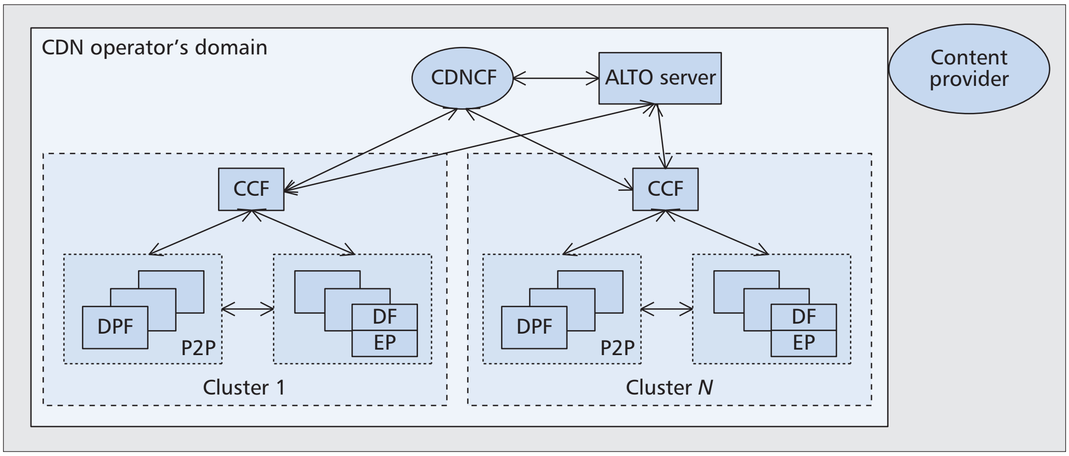

Figure 1. *Operator-owned CDN.*

The large variety of contents, their rapidly increasing number and popularity dynamics, together with the diversification of service-level agreements with content providers all conspire to require a highly flexible content outsourcing strategy.

that takes into account a weighted combination of three metrics, namely inter-autonomous system (AS) distance, intra-AS distance, and end-to-end latency. Although this algorithm is flexible, the deployment of an agent in each replica server for metric measurement makes it complex and costly. Moreover, the active latency measurement techniques used by this algorithm introduce additional traffic to the Internet. As described in [11], Akamai uses a proprietary algorithm. It is a complex adaptive algorithm that takes into consideration a number of metrics such as replica server load, the reliability of loads between the client and each of the replica servers, and the bandwidth that is currently available to a cache server.

DESIGN GOALS

This section illustrates the objectives for the operator-owned CDN design with respect to the selected functionalities. In particular, specific design goals are derived from the challenges introduced earlier.

NETWORK-TAILORED CACHE PLACEMENT STRATEGIES

For CDNs of mobile network operators, the choice of cache server locations is tightly coupled to the availability of gateways or traffic breakout points. In order to enable resource saving, cache servers can only be placed where the content can be directly accessed in the network with minimal round-trip time between the users and the edge caches. This introduces a new variable compared to residential networks, as the cost of placing a cache server is associated with not only the storage and streaming equipment but also the traffic breakout cost. Optimal trade-off can only be achieved with models that take this issue into account. For this reason, our target cache placement strategy must most of all be tailored to the preexisting mobile networks.

FLEXIBLE CONTENT OUTSOURCING TECHNIQUES

The large variety of contents, their rapidly increasing number and popularity dynamics, together with the diversification of service-level agreements with content providers all conspire to require a highly flexible content outsourcing strategy. Flexibility refers to both hybrid push-pull paradigms and adaptive approaches that switch between these two fundamental schemes based on content popularity and feedback from content providers. A robust content fragmentation technique with appropriate granularity is a key element for enabling this flexibility.

BALANCED REPLICA PLACEMENT ALGORITHMS

In order to exploit the flexibility provided by the content outsourcing while addressing the challenge of cost reduction, fragments or full content replicas need to be pushed onto cache servers with a strategy that delivers a tailor-made trade-off between data transport costs and data storage/processing costs. Network cost awareness embedded in placement algorithms maximizes the efficiency of request routing but has to be balanced by the required caching resources. The appropriate balance depends on the cost models of the network operator resources, and the algorithm must be able to adapt to different cost ratios.

NETWORK-AWARE REQUEST ROUTING ALGORITHMS

Request routing takes place after and relies on the other CDN functions described above, and greatly impacts the CDN performance. The decentralization challenge introduced earlier translates into an architectural design where requests are handled locally and propagated toward centralized entities only when the requested content is not available. Furthermore, localization of request serving can only be achieved through the awareness of network characteristics such as the current path cost and delay as well as load on servers and network elements.

MOBILE-OPERATOR-OWNED CDN ARCHITECTURE

The hierarchical CDN architecture shown in Fig. 1 summarizes the goals described earlier. The architecture is inspired by the Telecommunications and Internet Converged Services and Protocols (TISPAN) CDN architecture [12], but it provides more decentralized request handling in order to allow integration with the mobile network architecture. Integration with the radio access network (RAN) for a mobile operator is not explicitly presented; our purpose is to introduce a high-level architecture to show the entities responsible for the baseline CDN functionalities, and present the hierarchical and cluster-based organization.

At the lowest level (i.e., the cache server), the outsourcing and delivery functions are decoupled to increase the scalability of the obtained solution. The distribution point function (DPF) is in charge of content storage, and the delivery function (DF) is in charge of content delivery; they both represent two logical functions of the cache server (in the remainder of the article, the term *cache server* will be used as a synonymous with the DPF and DF elements). The entry point (EP) function is responsible for redirecting users' requests (e.g., based on Domain Name Service, DNS, or HTTP redirection) to the DF. As a consequence of the proposed hierarchical decentralized architecture, we designed a bottom-up request routing strategy that tries to handle the requests of users locally first. The cluster control function (CCF) is in charge of managing the DPFs on a cluster. If a local-level cache miss occurs, the recursive nature of a bottom-up approach forwards the request to the higher level of the architecture and returns the CDN cluster that holds the content.

Control function elements (CDNCFs) are in charge of managing different clusters. If the content is not available in the overall CDN, it is retrieved from the origin server and cached in the DPF that first received the user request performing local on-demand caching. It ensures that the content is always present in the lowest level of the architecture to handle very specific local requests and avoid the volume of traffic through the core network. Figure 2 shows the logic exchange of messages during the DPF location process. Given the partial content retrieval strategy explained later, a content is composed of a prefix and a suffix. The initial segment of a content, made up of a number of chunks, is defined as prefix and the remainder chunks compose the suffix. If we take it as a hypothesis for the moment, we can identify two steps in the request routing strategy.

In the first step, the content request is forwarded to the CCF of the cluster to which the user belongs. At this point, four things can happen:

- Both the prefix and the suffix for the requested content are cached in the cluster.
- The prefix is cached in the cluster but the suffix is not (cache miss for the suffix).
- The prefix is not cached in the cluster but the suffix is (cache miss for the prefix).
- The content is not present in the cluster.

We decided to describe the second case to highlight the roles of the CCF and CDNCF during request routing; the reader can deduce by analogy what would happen in the other cases. The CCF needs to locate the cache server that holds the prefix. The label "net-aware" in the figure means that the CCF is acting as an application-layer traffic optimization (i.e., ALTO[1]) client to find the most suitable DPF that holds the prefix. In the second step, to solve the cache miss for the suffix, the user request is forwarded to the CDNCF. It finds the most suitable cluster holding the suffix with a request to the ALTO server. Once the cluster has been identified, the CCF responsible for it sends a request to the ALTO server to locate the cache that will start to forward the suffix to the user. The goal of the ALTO server is to provide guidance to the CDN, sorting the list of DPFs or clusters that store a requested content according to the network preference. Designing the proposed hierarchical architecture and the request routing strategy, we learned that a bottom-up approach can maximize the advantages of deep network caching and supports low-latency content delivery. Furthermore, the lowest-level CDN element is the first-line response to flash crowds. The impact of the flash-crowd problem is reduced given the smaller size of each highly distributed cluster.

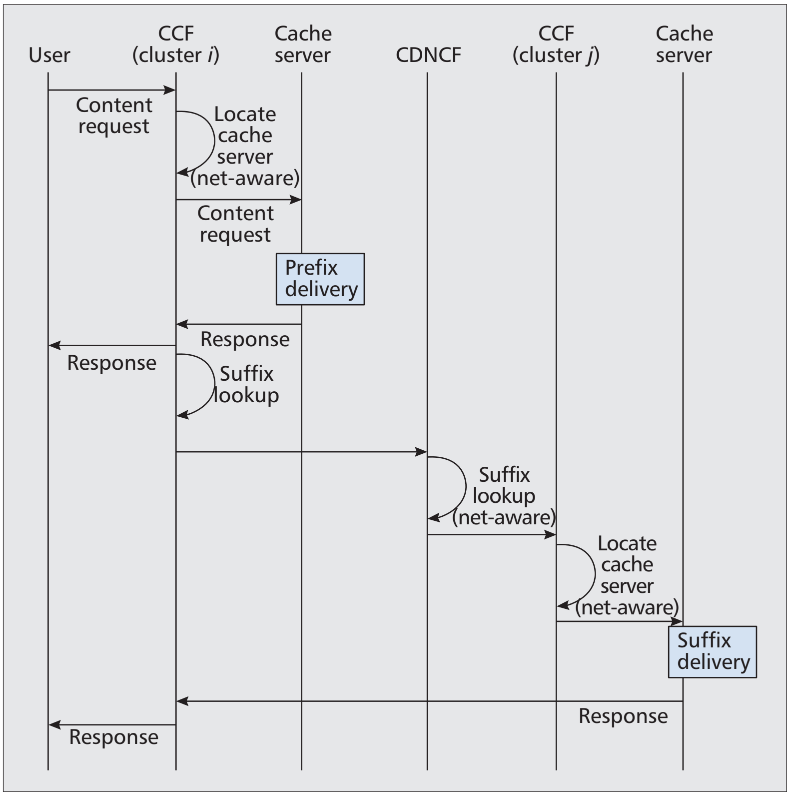

Figure 2. *CDN request routing.*

Beyond the request routing, CCF and CDNCF are responsible for all the baseline CDN functionalities. The CCF coordinates cluster-level content outsourcing and distribution, intracluster content directory (it acts as a tracker) and lookup, DPF selection upon initial request, and relocation. It also provides the first-line response to flash crowds by means of local content replication. The CDNCF coordinates global content outsourcing, intercluster content directory lookup, and reporting. Other entities in charge of authorization, accounting, content preparation, and content ingestion are out of the scope of this article.

---

[1] *http://datatracker.ietf.org/wg/alto/charter/*

DESIGN GUIDANCE

Referring to the baseline CDN functionalities of cache servers placement, content outsourcing, and replica placement, we describe the adopted solutions in the design of our operator-owned CDN.

CACHE SERVER PLACEMENT

The motivation to reduce traffic in the mobile operator core network and to keep content delivery paths to the user equipment short induces placement of the CDN entry points (i.e., EPs) and cache servers closer to a requesting user in the network topology, for example, in the *backhaul network*, which connects the RAN with the core network, or even in the RAN. Such placement of cache servers assumes network capability to route or switch data packets from the cache server to the user mobile device on the shortest path. As depicted in Fig. 3, mobile carrier networks rely on the PDN GW to anchor a mobile device IP address, which is used for data communication and routing of associated packets. All data traffic to and from a mobile device traverses the assigned PDN GW, and is tunneled between the PDN GW and the mobile device's current location. Nowadays, PDN GWs are typically deployed in centralized locations, close to the mobile carrier Internet exchange point by means of the operator network's connection to the public Internet. Solely the distribution of content cache servers still requires the mobile device's data traffic to traverse the associated central PDN GW, which results in inefficient and costly routes. Placement of a content cache server at the topology level of PDN GWs (Fig. 3, cache position L_0) can reduce traffic that traverses the router(s) providing the Internet exchange point(s), but does not reduce the traffic volume in the core network. Current standards activity in the Third Generation Partnership Project (3GPP) on technology for SIPTO enables traffic breakout of users' data flows below the level of centralized PDN GWs using L-GWs [3]. As does the PDN GW, the L-GWs serve as anchors for mobile device IP addresses. L-GWs can be distributed throughout

The motivation to reduce traffic in the mobile operator core network and to keep content delivery paths to the user equipment short induces placement of the CDN entry points (i.e. EP) and cache servers closer to a requesting user in the network topology.

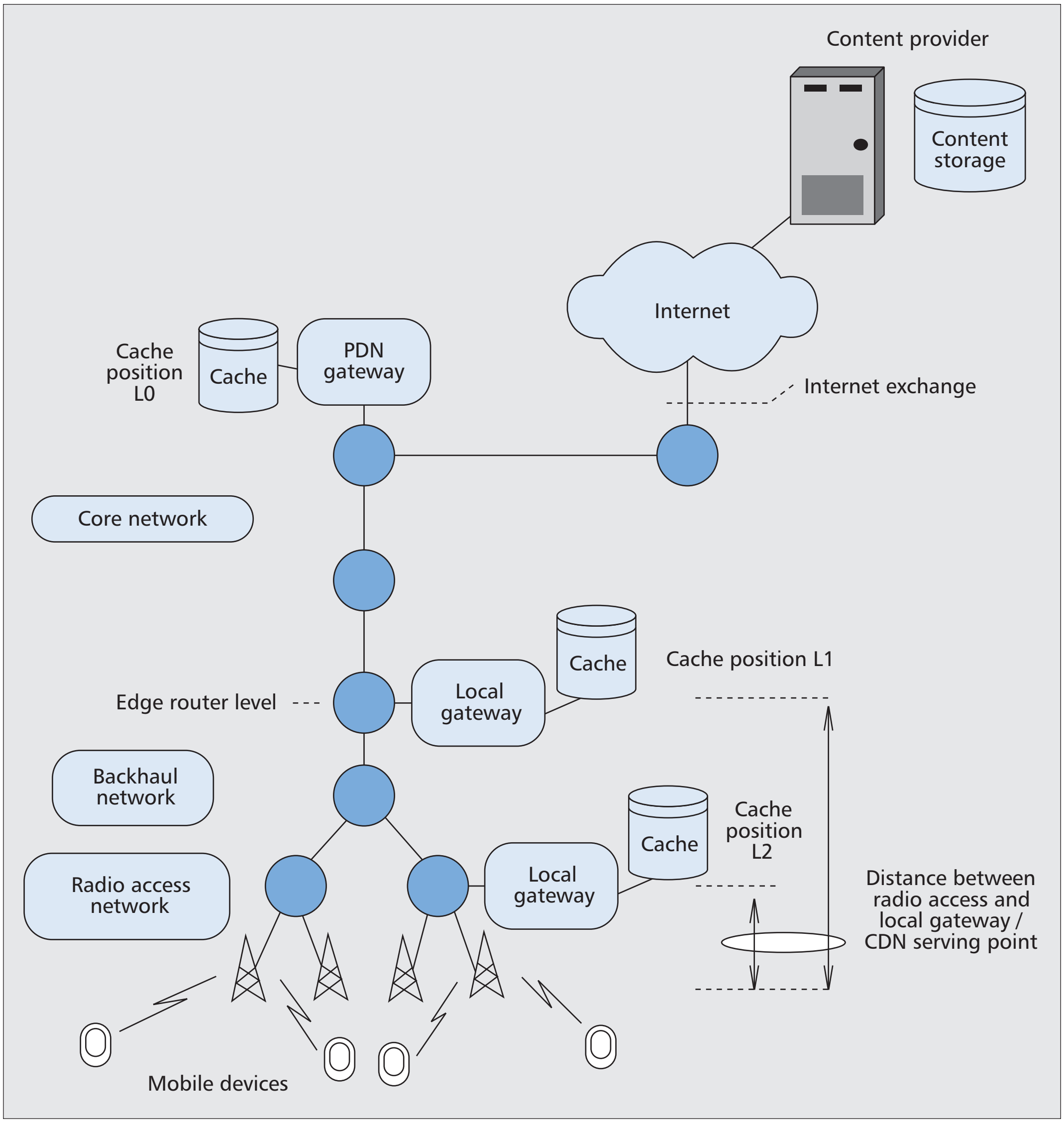

Figure 3. *General structure of a mobile carrier network and enabled access to distributed content cache servers by means of traffic breakout.*

the network and are assigned according to an attaching mobile device location, which keeps the mandatory delivery path through the tunnel between the mobile device and the topological anchor of its IP address short. Placement of content cache servers close to distributed L-GWs allows reducing the path and the associated transport costs for the delivery of cached content to a mobile device, as depicted in Fig. 3 (cache position L_1).

Backhaul networks, which still have asynchronous transfer mode (ATM) and time-division multiplexing (TDM) technologies inherited from the second generation of mobile networks, are about to turn into pure packet-based transport networks and rely typically on switching technologies,such as multiprotocol label switching (MPLS). IP routing of data packets is considered only above MPLS ingress/egress routers, which represent the endpoints of an MPLS tunnel. This network level is denoted as the *edge router* level of the core network.

Placing L-GWs, EPs, and cache servers below edge router level (Fig. 3, cache position L_2) may be disadvantageous for the following reason: a DF may not find requested content on the local cache server, but can retrieve the content from a remote cache server. In such a case, the path from the remote cache server up in the network topology to the edge router through the switched backhaul network and then down again to the user DF can result in higher network costs compared to a setup that places breakout points and cache servers on the edge router level. Even though the DF has been moved closer to the requesting user, in the case of frequent delivery from remote caches such optimized placement may not pay off due to the transport network topology in the backhaul and the resulting suboptimal delivery path.

Figure 4 illustrates the results from a computation of content retrieval costs for different cache hit rates and different cache server positions obtained by analytical evaluation. The evaluation focuses on transport costs, which depend on the topological distance between a delivering cache server and the associated client. The cache hit rate is a measure of the percentage of the requested content that was already cached in the CDN. The chosen model assumes that for distance $d_{PSGW} = 10$, the PDN GW and the CDN caches are centralized in the core network and close to a cache server or Internet Exchange (i.e. Cache Position L_0), whereas for $d_{PSGW} = 0$, an L-GW and DF are on the same topology level as the radio access network (i.e., cache position L_2). At distance $d_{PSGW} = 2$, the model considers the edge router level (i.e., cache position L_1), the lowest level in the topology where data packets can be routed between backhaul network branches. According to Fig. 4, costs have been normalized to the maximum costs obtained in a centralized setup.

A thorough analysis based on different cost models has shown that if the caches are placed in a centralized position, the gain of increasing the cached content (i.e., cache hit rates) is negligible because the load crosses the core operator network and negatively impacts the total cost. On the contrary, the placement of L-GWs, EPs, and cache servers at the edge router level represents a suitable trade-off between cost reduction and distance between users and EPs. This topology level allows the shortest distances for intercache routing while caches are reasonably close to users. Figure 4 shows that content retrieval costs can drop by more than 50 percent when 80 percent of the requested content is cached by the CDN, whereas the remaining requests are served from a central cache server.

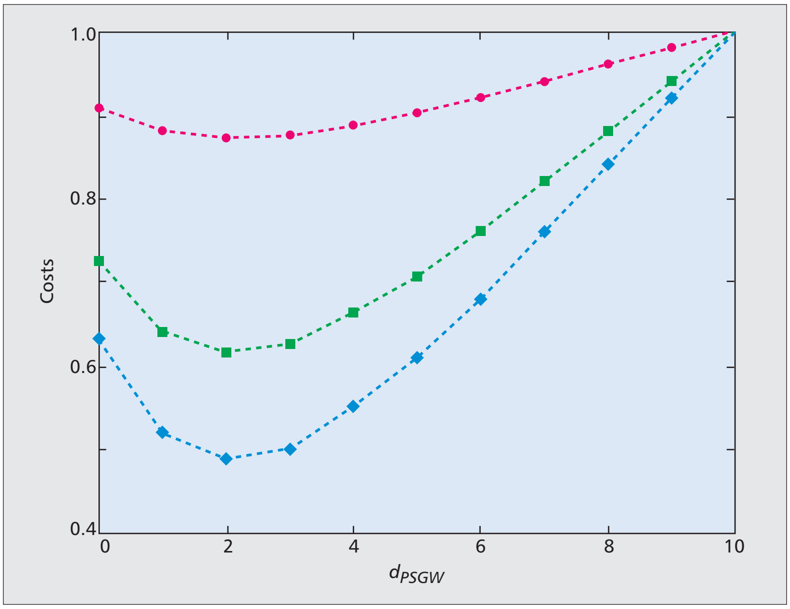

Figure 4. *Normalized content retrieval costs for different cache server locations (d_{PSGW}) and different cache hit rates (20% [circles], 60% [squares] and 80% [diamonds]).*

The analysis also shows that mobile operators can further increase the gain in cost reduction if they enable traffic breakout in the switched backhaul network to efficiently access CDN cache servers located on the same level. All the following strategies to complete the CDN design have to cope with the level of cache decentralization, as we learned from this study for a mobile network operator.

CONTENT OUTSOURCING AND RETRIEVAL

The trade-off to take into account in the design of the content outsourcing strategy is between the storage space in the cache servers and the stream setup latency (we refer to this as *latency* in the remainder of the article) experienced by users during content retrieval. Storage space is an important issue when dealing with the highly distributed architecture that characterizes a mobile operator. To obtain such a trade-off, a pure push-based or pull-based content outsourcing strategy is each unsuitable: the first is not cost effective for the storage, and the second negatively affects the latency perceived by users. Moreover, as learned earlier, it is not cost-wise for the operator to retrieve the content from the core network if a cache miss occurs (i.e., the content is not available in the cache server). To reduce the load that crosses the core network, decrease the latency, load balance the retrieval process, and quickly adapt to overload, our simulation results point toward the preference of a

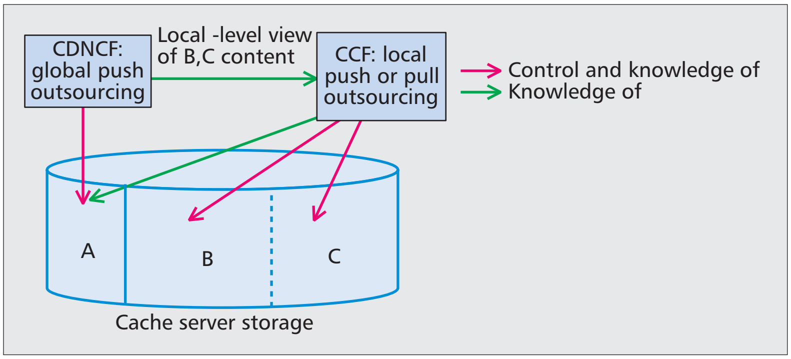

Figure 5. *Logical storage space.*

collaborative BitTorrent-like scheme among the cache servers. Furthermore, the chunk-based nature of the content inside the CDN allows its flexible management and the application of a partial caching technique. The idea of the partial caching technique is to proactively push an initial segment of a content made up of a number of chunks (i.e., the prefix) and retrieve the remaining chunks (i.e., the suffix) through the overlay P2P network of the cache servers managed by a tracker or directly from the origin server using an on demand pull strategy. The size of the prefix depends on the trade-off between the cache servers' memory space and the need to shield users from delay and loss along the path on the overlay network.

At this point, a further step toward the users' quality of experience requirement is done to ensure fast retrieval for those kinds of content with high quality of service (QoS) constraints, low dynamics, and high availability requirements. For so-called *Class A* content, a proactive push-based outsourcing strategy takes place, and the entire content is pushed to the cache servers.

Considering our highly distributed cache servers placement scheme, we have to assume that there are contents of global interest and others dealing with local/regional interest. For this reason, we introduce the concept of *global* and *local* push outsourcing strategies. Class A contents are supposed to reach a global audience; in other words, each cache server of the network needs to be aware of their position (through the tracker). On the contrary, contents that address primarily a local interest, categorized as *Class B* contents, can be proactively pushed only to relevant cache servers in the network, and their visibility can be limited to the region of interest.

The remaining contents, which require lower QoS or are characterized by high dynamics, are best cached on demand upon the first request using a pull strategy, and is called *Class C* content herein.

Figure 5 visually shows the logical organization of a cache server and the content-based outsourcing strategies. The vertical dashed line represents the dynamic space allocation for Class B and Class C content controlled by the CCF. Depending on users' demand, the CCF flexibly allocates more cache space to one or the other content class. In our simulation study, omitted for the sake of brevity, we evaluated the cache hit ratio and retrieval latency considering all the described outsourcing strategies. As we learned from our simulation results, the proactive push of partial or complete content produces a performance improvement only if it is coupled with a frequent replica placement algorithm. If the replica placement algorithm does not follow the change of popularity of the content, events of cache pollution (i.e., the cache server is full of stale content) will happen, and the cache hit ratio will decrease.

In general, to increase the cache hit rate, it could seem more reasonable to proactively push the Class A content and retrieve all the others on demand. To do so, the retrieval latency for the Class B content is negatively affected, even though the effect is mitigated by the collaborative scheme among the cache servers.

As a conclusion, we learned that the content outsourcing needs to be flexible to cope with the decentralized CDN. Moreover, a content-based outsourcing strategy combined with a cooperative multisource scheme among the cache servers can reduce the load on the core network, balance the storage size, and reduce the latency for retrieval. The more the replica placement algorithm is adaptive to the changes of popularity, the more effective the proactive outsourcing becomes.

REPLICA PLACEMENT

The problem of replica placement consists of finding an assignment (X) of object replicas to a set of cache servers such that a certain metric is maximized. Although the heuristics presented earlier are based on coarse-grained metrics inferred by third parties, in our approach we aimed at exploiting the network knowledge of the operator that owns the CDN. In this section we outline our approach without detailing the actual algorithm and point out the key lessons learned.

The first step in the replica placement design was to identify the cost factors for the network/CDN operator and their ratios. We expressed the total cost as the sum of the data transport costs, or network cost, $C_{NW}(X)$, and the cost of caching the content, $C_C(X)$, which contains the costs of physical storage and processing power consumed by a cache server to serve content requests. These costs include both capital and operational expenditures.

$$C(X) = C_{NW}(X) + C_C(X). \quad (1)$$

It is possible to express these costs as a function of the transferred or served data unit (we omit the full formula for the sake of brevity). A key observation is that the ratio between the cost per unit of transmitted data and the cost per unit of stored data is a fundamental factor that determines the optimal trade-off between data transport and data caching costs. Consequently, the replica placement algorithm needs to be parameterized to this factor, as it varies across operators depending on their internal cost model. For the purpose of defining the algorithm, we applied a wide spectrum of values centered around commercial prices of reference cloud storage and processing services. While these prices are also influenced by market factors, the cost ratio turned out to be similar across different service providers. We concluded that the ratio is more affected by internal operator costs than its marketing strategy.

Another lesson learned through expressing Eq. 1 in a closed form is that leased transport capacity, due to its fixed or step-like costs, cannot be expressed as a function of the actual data volume. Hence, we modeled these costs by adopting a demand-based approach, where we assigned each backhaul link a unit network cost that is proportional to the number of subscribers served by that link.

We observed that the Eq. 1 solution space grows exponentially with the number of contents and the number of cache servers. Brute force is therefore inapplicable; only suboptimal solutions can be found through (meta-)heuristics. We chose to break the problem down into two steps related to the two variables, network cost and storage cost. In the first step, we sample the solution space with respect to one of the variables in order to set a target cost, while in the second step we apply greedy heuristics within the restricted solution space identified in the first step. We chose to apply network cost in the first step and caching cost in the second step, which led us to call our approach the "network cost-driven greedy-like replica placement algorithm."

As a benchmark, we compared our algorithm with a network cost-unaware reference algorithm that randomly places a number of content replicas proportional to its popularity.

We evaluated the performance in terms of aggregated network cost defined as the cost accumulated by one content chunk during its delivery from the cache server to the user. The results highlighted that the network-aware algorithm, due to its popularity bias, is sensitive to the accuracy of the popularity estimation. During the simulation time (i.e., 6 h) each content experiences all the phases of its popularity:

- An initial phase where it acquires interest
- A peak
- A decrease in the number of requests

The simulated contents (i.e., 5000 objects divided into standard and high definition in an equal ratio) experience their peak of popularity in different time intervals; this assumption introduces a large challenge for replica placement. We chose these extremely variable simulation conditions on purpose to evaluate our algorithm in the worst case scenario.

Our simulation results showed that if the network-aware replica placement is performed with a periodic interval of 1 h (i.e., six times over the total simulation time), there are no advantages compared to the network-unaware algorithm. On the contrary, for a 15 min placement interval, our algorithm reduces network cost by 50 percent compared to the reference algorithm.

The algorithm we defined is the result of several observations aimed at finding a trade-off between algorithm complexity and near optimality. In particular, we learned that in applications where network cost requirements outweigh storage cost ones, it is beneficial to provide a target network cost that will not be exceeded assuming that the popularity estimation is correct. This allows network operators to perform more accurate cost planning.

Design goal	Design principle
Network-tailored cache placement	–Placement of L-GWs, EPs, and cache servers at the edge router level
Flexible content outsourcing	–Chunk-based nature of content –BitTorrent-like scheme –Partial content outsourcing –Pull and push outsourcing –Local (on a cluster by the CCF) and global (on the CDN by the CDNCF) outsourcing
Balanced replica placement	–Two-step algorithm based on the operator's network knowledge (network cost and storage cost as variables)
Network-aware request routing	–Decentralized, hierarchical, cluster-based CDN organization –Interactions with an ALTO server

Table 1. *Design goals/principles summary.*

CONCLUSIONS AND LESSONS LEARNED

Recently, we have observed a rapid rise in the popularity of video services for mobile terminals. Since video delivery is very bandwidth-intensive and requires high data rates, mobile operators have started to study solutions to optimize network resources. Among these, one of the most interesting approaches is based on the extension of traditional CDNs by distributing cache servers close to users. This approach permits reducing the amount of traffic load on the backhaul and core network, and the latency experienced by users.

The decentralized nature of the mobile operator CDN and the output of the study on cache server placement presented earlier have driven many of the design steps. To minimize the cache storage space and address the peculiar nature of the content given by the high decentralization, the flexible combination of proactive and on-demand partial outsourcing strategies based on content characteristics have been presented. Furthermore, the introduction of a collaborative communication scheme among cache servers permitsthe volume of traffic across the core network to be reduced. The simulation study of the proposed outsourcing strategy highlighted the importance of a well designed replica placement algorithm to effectively utilize the cache servers' storage and make partial caching advantageous.

The key functionality of our replica placement algorithm has been the network awareness that characterizes an operator-owned CDN and guarantees flexible cost planning for the operator, which can drive the replica placement algorithm based on its own policies and priorities. The network-aware request routing as a consequence of the proposed hierarchical architecture enforced the advantages of the presented replica placement. The hierarchical architecture described earlier highlighted how the discussed design guidance can be applied and presented the functional entities responsible for the differ-

The key functionality of our replica placement algorithm has been the network awareness that characterizes an operator-owned CDN and guarantees flexible cost planning for the operator that can drive the replica placement algorithm based on its own policies and priorities.

ent features of the operator-owned highly distributed CDN. Table 1 helps to summarize the design goals discussed in this article and the proposed solutions to achieve them.

ACKNOWLEDGMENTS

This publication is based on work performed in the framework of the Project COAST-ICT-248036, which is partially funded by the European Community.

REFERENCES

[1] G. Peng, "CDN: Content Distribution Network," CoRR, vol. cs.NI/0411069, 2004.

[2] A. Vakali and G. Pallis, "Content Delivery Networks: Status and Trends," *IEEE Internet Computing*, vol. 7, 2003, pp. 68–74.

[3] "3GPP General Packet Radio Service (GPRS) Enhancements for Evolved Universal Terrestrial Radio Access Network E-UTRAN access," July 2011. 3GPP TS 23.401 v. 8.3.0.

[4] S. Jamin *et al.*, "On the Placement of Internet Instrumentation," *IEEE INFOCOM*, Mar. 2000.

[5] Y. Bartal, "Probabilistic Approximation of Metric Spaces and Its Algorithmic Applications," *37th Annual Symp. Foundations of Computer Science*, 1996, pp. 184–93.

[6] E. Cronin *et al.*, "Constrained Mirror Placement on the Internet," *IEEE JSAC*, 2002, pp. 31–40.

[7] K. Johnson *et al.*, "The Measured Performance of Content Distribution Networks," *Comp. Commun.*, vol. 24, Feb. 2001, pp. 202–06.

[8] M. J. Freedman, E. Freudenthal, and D. Mazieres, "Democratizing Content Publication with Coral," *Proc. 1st Conf. Symp. Networked Systems Design and Implementation*, vol. 1, Berkeley, CA, 2004, pp. 18–18.

[9] S. S. H. Tse, "Approximate Algorithms for Document Placement in Distributed Web Servers," *IEEE Trans. Parallel Distrib. Sys.*, vol. 16, June 2005, pp. 489–96.

[10] G. Pallis *et al.*, "A Latency-Based Object Placement Approach in Content Distribution Networks," *Proc. Third Latin American Web Congress*, Washington, DC, 2005, pp. 140–.

[11] J. Dilley *et al.*, "Globally Distributed Content Delivery," *IEEE Internet Computing*, vol. 6, 2002, pp. 50– 58.

[12] ETSI, "Telecommunications and Internet Converged Services and Protocols for Advanced Networking (TISPAN); Content Delivery Network (CDN) Architecture Interconnection with TISPAN IPTV Architectures," TS 182 019 V3.1.1.

BIOGRAPHIES

STELLA SPAGNA (stella.spagna@iet.unipi.it) received her Laurea degree in telecommunication engineering cum laude in September 2008 from the University of Pisa, Italy. In 2012 she received her Ph.D. degree in information engineering from the same university, discussing the thesis *Congestion Avoidance and Traffic Optimization in Distributed Systems*.

MARCO LIEBSCH (marco.liebsch@neclab.eu) is currently working as a senior researcher at NEC Laboratories Europe in the area of mobility management, mobile content distribution, and mobile cloud networking. He has worked on different EU research projects and is contributing to standards in the IETF and 3GPP. For his thesis, *Paging and Power Saving in IP-Based Mobile Communication Networks*, he received a Ph.D. from the University of Karlsruhe, Germany, in 2007. He has a long record of IETF contributions as well as RFC, journal, and conference publications.

ROBERTO BALDESSARI (roberto.baldessari@neclab.eu) received his Master's degree in telecommunications engineering from the University of Padua. After working for Telecom Italia Laboratories on IPv6 mobility in Turin, Italy, he joined NEC Laboratories Europe where he currently leads the Intelligent Transport Systems (ITS) Group. In the past nine years, he has been carrying out research, standardization and business promotion activities in the areas of IPv6, network mobility, mobile content delivery, vehicular communications and transport optimization.

SAVERIO NICCOLINI (saverio.niccolini@neclab.eu) leads a research group focusing on the areas of network traffic monitoring, service analytics, and software-defined networking inside the network research division of NEC Laboratories Europe, Heidelberg, Germany. He received an M.S. degree in telecommunications engineering and a Ph.D. degree in information engineering from the University of Pisa in 2001 and 2004, respectively.

STEFAN SCHMID (stefan.schmid@neclab.eu) received his Ph.D. in computer science from Lancaster University, United Kingdom, in 2002. He then joined NEC Laboratories Europe, where he was actively involved in EU projects in the areas of 4G and future Internet research. In 2007, he was appointed manager of the Next Generation Networking research group. Since 2009 he has led the mobile and wireless networks research at NEC Laboratories. He has published more than 60 scientific papers in international workshops, conferences, and journals.

ROSARIO G. GARROPPO (rosario.garroppo@iet.unipi.it) is an assistant professor at the University of Pisa and member of IEEE ComSoc. His research interests include experimental measurements and traffic modeling in broadband networks, traffic control techniques for multimedia services, green networking, and QoS guarantee in wireless networks. He has published more than 90 papers in international journals and conferences.

KAZUNORI OZAWA's biography was unavailable at the time this article was published.

JUN AWANO's biography was unavailable at the time this article was published.