On Private CDNs with Off-Sourced Network Infrastructures: a Model and a Case Study

Claudia Canali, Andrea Corbelli, and Riccardo Lancellotti

***Abstract*—The delivery of multimedia contents through a Content Delivery Network (CDN) is typically handled by a specific third party, separated from the content provider. However, in some specific cases, the content provider may be interested in carrying out this function using a Private CDN, possibly using an off-sourced network infrastructure. This scenario poses new challenges and limitations with respect to the typical case of content delivery. First, the systems has to face a different workload as the content consumer are typically part of the same organization that is the content provider. Second, the off-sourced nature of the network infrastructure has a major impact on the available choices for CDN design. In this paper we develop an exact mathematical model for the design of a Private CDN addressing the issues and the constraints typical of such scenario. Furthermore, we analyze different heuristics to solve the optimization problem. We apply the proposed model to a real case study and validate the results by means of simulation.**

***Index Terms*—CDN, Multimedia contents, Case study.**

I. Introduction

Content Delivery Networks (CDNs) have gained an increasing popularity over the years as dedicated infrastructures for the delivery of multimedia streaming contents. The typical structure of a CDN is represented in Figure 1 and consists of a network distributed infrastructure, where the servers (namely *edge servers*) receive content from a centralized *origin server* in the form of a media stream and forward it to the clients. The edge servers derive their names from the placement that is typically on the *network edge* (as close as possible to the clients) to reduce latency and improve the performance of media content delivery. Furthermore, the edge servers can take advantage of their position and act as demultiplexer, that is splitting one media stream into a separate stream for each client, with evident benefits for these infrastructures due to the reduced risk of network congestion and consequent poor performance. Finally, each edge server can perform *transcoding* (this function is represented as a star-shaped object on the edge servers in Figure 1). The transcoding function adapts the quality of the stream to the specific characteristics of a client in terms of encoding, bandwidth and resolution (as shown by the different types of objects sent to the clients in figure).

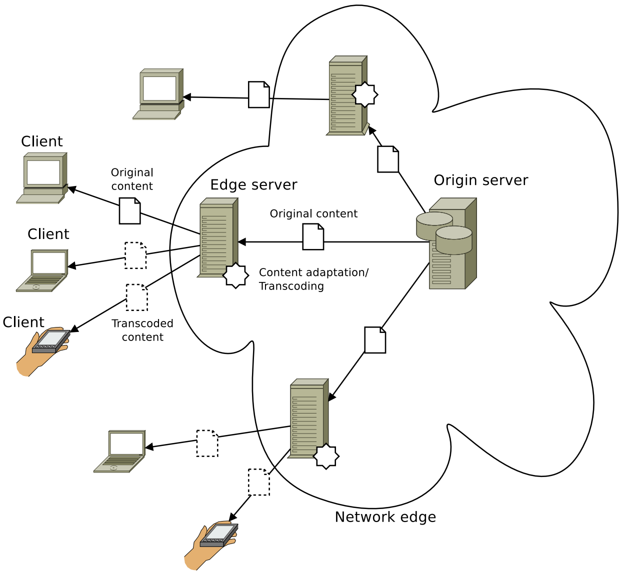


Fig. 1. CDN infrastructure

Several studies exist in literature proposing strategies and models for edge server placement in traditional highly distributed CDN infrastructures, such as those owned by providers like Akamai [1] and Limelight [2]: these studies aim at achieving the best possible placement of the edge servers in order to satisfy the end-users quality of service, the requirements about bandwidth and latency, as well as the need for the CDN provider to minimize the infrastructure costs [3]–[6]. However, all these results are applied to traditional CDNs, where a CDN provider offers services to a multitude of content providers and where the CDN provider has no constraints on the placement of the edge servers. Little effort has been devoted to the case of a *Private CDN (P-CDN)* where a company that already has an access to a geographic network infrastructure (typically leased from a network operator that remains the maintainer of the infrastructure) wants to deliver content to its branches placed in geographically distinct locations.

The case of the design of a P-CDN presents peculiarities that have major impact on the expected workload and on the constraints about the placement of edge servers, arising new issues with respect to the design of a typical CDN infrastructure. First of all, the presence of just one content provider, which is the only customer of the network infrastructure, results in a very different workload with respect to the typical case where many clients access content from multiple providers. Furthermore, the outsourcing of the network infrastructure to a network provider poses major constraints in the design of the

Manuscript received August, 2018; revised October 22, 2018. Date of publication December 19, 2018. The associate editor Prof. Nikola Rožić has been coordinating the review of this manuscript and approved it for publication.

Authors are with the Department of Engineering “Enzo Ferrari”, University of Modena and Reggio Emilia, Italy. E-mails: {claudia.canali, andrea.corbelli, riccardo.lancellotti}@unimore.it

Digital Object Identifier (DOI): 10.24138/jcomss.v14i4.607

P-CDN in terms of the possibility to place the edge servers for the delivery of the multimedia streaming contents. This different scenario motivates our effort to propose innovative solutions specifically tailored for a P-CDN.

The main contribution of this paper is the proposal of an exact mathematical model for the design of a P-CDN addressing the issues and the constraints typical of such scenario. An additional contribution of our proposal is introducing the concept of Satellite Locations to consider areas not covered by an edge server. Moreover, we propose and compare different heuristics to solve the underlying optimization problem. As a final qualifying point of our research, we present the description of an industrial experience based on a private company in need of deploying a P-CDN where the networking infrastructure is outsourced to a third party. The industrial reference scenario used in our case study depicts a large Italian company aiming at deploying a nation-wide P-CDN. The proposed model is applied to this real-world case study and is validated by means of a simulator. The simulation results show how the presence of a P-CDN can significantly increases (by more than eight times) the number of clients accessing the multimedia data at the highest quality. Furthermore, our results evidence a major trade-off between the number of used edge servers and the media quality offered by the CDN to the end users. Finally, it is worth to note that a critical parameter for tuning the model is related to the decision about a company branch being large enough to be chosen to host an edge server; this criticality is motivated by the cost of edge servers that may be high for a single content provider. A preliminary version of this paper was proposed in [7]; however, this paper is a clear step ahead with respect to that first draft, proposing a more complete discussion of the CDN model and providing additional experimental results.

The remainder of this paper is organized as follows. Section II discusses the related work and positions our contribution with respect to the state of the art. Section III describes our model for placing the edge servers and proposes the heuristics to solve the optimization problem. Section IV describes the case study considered to validate our model. Finally, Section V concludes the paper with some final remarks.

II. Related Work

Over the years the attention of many researchers has been devoted to Content Delivery Networks (CDNs) as these systems gained popularity in the field of content distribution. The main research trends about content delivery networks (CDNs) and user access paradigms are analyzed and presented in [3]; on the other hand, the study in [4] focuses specifically on the delivery of multimedia content through CDNs.

A critical issue in CDN systems development is the choice of the edge servers placement that should minimize the infrastructure costs for the provider while, at the same time, satisfying several requirements about bandwidth, latency, and quality of service for the end-users. A comprehensive survey on edge server placement algorithms in traditional and emerging paradigm-based CDNs is presented in [5]. Another study that deserves to be mentioned in this field is [6], that provides design principles for a highly distributed CDN while focusing on four key aspects: the optimal location of edge servers with caching functionalities, mechanisms for request routing, content replica placement, and content outsourcing and retrieval. Our proposal fits in this research field, and concerns an optimization model to minimize the number of edge servers used in a live-streaming scenario for multimedia contents delivery, where the edge servers are able to perform online transcoding [8], [9]. A significant difference between our proposal and the majority of the state-of-the-art studies is that they focus on the scenario of highly distributed commercial CDN infrastructures, owned by CDN owners such as Akamai [1] and Limelight Networks [2], while in our experience paper the focus is about a P-CDN, that is owned and used by a company to serve its own contents, with the networking infrastructure outsourced to a third party. It is worth to consider that the challenges presented by this scenario are different from those of a typical CDN. For example, in this case the content provider represents the only customer of the network infrastructure, hence the resulting workload is quite different from the typical case where many clients access contents from multiple providers. Furthermore, the fact that the network infrastructure is off-sourced to an external network provider poses major constraints on the possibility to place the edge servers of the infrastructure.

Other studies in literature focus on Telco-CDNs, where the content distribution services are managed by telecommunications service providers (TSPs) that began to launch their own content delivery networks as a mean to lessen the demand on the network backbone and to reduce infrastructure investments. In this case the network operator is able to manage the complete system (it controls both the infrastructure and the content delivery overlay), so that networking resources are optimally utilized. Among these studies, we cite [10], that proposes an algorithm for the placement of video chunks on the edge servers, and [11], where an architecture for on-demand service deployment over a Telco-CDN is presented. Also in the specific case considered in our study the owner of the CDN is a telecommunication company, however the network infrastructure is outsourced to an external network provider, with the consequence that the edge servers can not be arbitrarily located but should be placed in determined spots corresponding to the company branches.

As regards the scalability of the origin server, it is not the focus of this paper since many techniques already exist to ensure such property. The origin server can be, indeed, implemented as a distributed data center where each server hosts several virtual machines (VMs), and optimization techniques may be exploited to distribute the load among the available resources. For example, in [12] the authors present a mechanism to automatically cluster similar VMs in order to guarantee the scalability of a distributed data center monitoring and management.

A related field of investigation concerns the distribution of video and multimedia, usually accessed through mobile devices in the form of streaming video and online applications: in this context, existing studies are typically focused on improving the performance of content distribution on mobile networks [13], [14]. Even if mobile users may be included in our scenario, they are local clients on company LAN locations that access contents using mobile devices connected through a WiFi access point. In this case, the company represents the only provider that delivers contents through the network infrastructure. For this reason, the highly transient properties of mobile devices and changing locations of the users accessing contents from multiple providers do not represent an issue. Instead, we may focus on the challenge of identifying the best locations to place the edge servers among the company branches geographical positions in order to optimize the performance of the P-CDN.

III. P-CDN Model

In this section we will present the P-CDN problem, focusing on:

- introducing the model of a P-CDN, both in general case and in the case where we are using the infrastructure for live streaming
- defining the underlying mathematical problem
- outlining some heuristics used to solve the problem

A. General Scenario Definition

In the general scenario of a P-CDN, we assume the network to be characterized by *locations*, corresponding to the branches of the company deploying the P-CDN (and that are hosting the clients). These locations are denoted as $l \in \mathbf{L}$ in the remaining of the paper. We characterize each location with a download and an upload bandwidth, that we denote as BWd_l and BWu_l, respectively. Furthermore, we define BWo as the bandwidth available at the origin server for feeding the P-CDN. In our analysis we assume that the network is not used just for the CDN operations but may have additional background traffic. As a consequence, the considered bandwidth is not the nominal available bandwidth of the links but is an estimate of the actually unused bandwidth. The CDN content is consumed by a set of heterogeneous clients, where each type of client may require a specific stream characterized by a type of encoding $t \in \mathbf{T}$ and by a bandwidth requirement BW_t. Let $\mathbf{C}_{t,l}$ be the set of clients in location l that consume a stream with encoding t. As the CDN is used mainly on working hours, we may assume that not every user (and, as a consequence, not every client) will actually watch the video stream. For this reason we model our scenario in such a way that just a fraction of the total clients will actually watch the video stream. In the following of our analysis, the set $\mathbf{C}_{t,l}$ considered in the model consists of an estimated set of clients that will likely concurrently access the live stream.

In the absence of a CDN, we consider that the delivery of multimedia contents involves just the origin server and the clients. If multicast is enabled throughout the network, the content delivery is straightforward as we can simply provide one stream to a different IP multicast address for every considered encoding $t \in \mathbf{T}$. In the reference scenario, multicast can be enabled in each specific location requesting this feature to the network operator. Specifically, in a generic location l, we denote $M_l = 1$ if multicast is enabled for that location and $M_l = 0$ otherwise (implicitly we define also a variable $\overline{M_l} = \neg M_l$ that is the boolean negation of variable M_l). We anticipate that multicast has not been widely explored in our scenario for a twofold reason. First, the economic requests from the network operator for enabling multicast in each location are quite high. Second, enabling multicast support in each location requires the installation (and maintenance) of additional software at the level of clients. For these reasons we consider multicast just as a fallback choice.

TABLE I
Model Notation

Symbol	Meaning
$\mathbf{T}$	Set of encoding types
$\mathbf{L}$	Set of locations (company branches)
$\mathbf{L_e}$	Set of locations with edge server
$\mathbf{C}_l$	Set of client at location l
$\mathbf{C}_{t,l}$	Set of client at location l requiring encoding t
BW_t	Bandwidth for encoding t
BWd_l	Download bandwidth at location l
BWu_l	Upload bandwidth at location l
BW_l'	bandwidth requirement at location l
BWo	Bandwidth at origin server
M_l	Flag for multicast at location l
E_l	Flag for edge server at location l
P_m	Cost of enabling multicast in a location
P_e	Cost of installing an edge server in a location

The main approach for multimedia stream delivery is to introduce a group of *edge servers* that can be located in the company branches to act as content distributors for the nearby clients. Let $\mathbf{L_e}$ be the set of locations that host an edge server. In our model, we assume that each edge server may also implement a *transcoding* function. With transcoding, a single high quality stream can be used to generate lower-quality streams. We model the presence of edge servers using a boolean variable to decide if a location should host or not an edge server. Specifically, E_l has value 1 if in location l we have an edge node (that is if $l \in \mathbf{L_e}$), 0 otherwise. Again, we also define a set of variables $\overline{E_l} = \neg E_l, \forall l \in \mathbf{L}$ that are the boolean negation of variables E_l.

To make the model more readable, we summarize the main variables and parameters of the model in Table I.

B. Live-Streaming Scenario

In order to model a live streaming scenario, it is useful to introduce a detailed model of the bandwidth consumption at the level of each location $l \in \mathbf{L}$. Specifically, the bandwidth will depend on whether we have a transcoding edge server, we use multicast or nothing is used (the case where we have both multicast and an edge server is not considered as it would make no sense). Let BW_l' be the bandwidth requirement for clients at location l. Recalling that the variables M_l and E_l are used to model the presence of multicast or edge server, we can define the bandwidth requirement:

$$BW_l' = \begin{cases} \sum_{t \in \mathbf{T}} BW_t \cdot |\mathbf{C}_{t,l}|, & \text{if } M_l = 0, E_l = 0 \\ \sum_{t \in \mathbf{T}} BW_t & \text{if } M_l = 1, E_l = 0 \\ \max(BW_t, t \in \mathbf{T}) & \text{if } M_l = 0, E_l = 1 \end{cases} \quad (1)$$

Or, as a more compact form we can write:

$$\begin{aligned} BW_l' =& \overline{M_l} \cdot \overline{E_l} \cdot \sum_{t\in\mathbf{T}} BW_t \cdot |\mathbf{C}_{t,l}| + \\ & M_l \cdot \overline{E_l} \cdot \sum_{t\in\mathbf{T}} BW_t + \\ & \overline{M_l} \cdot E_l \cdot \max(BW_t, t \in \mathbf{T}) \end{aligned} \tag{2}$$

Before delving into the model details it is important to point out that we assume an edge server as being able support only the clients in the same location as the server. More complex scenarios, where a single edge server can serve clients in more than one location will be analyzed in the following. The optimization problem of the live streaming scenario can be modeled as a problem of minimizing the infrastructure cost. If P_m is the cost of multicast and P_e is the cost of an edge server, we can write:

$$\min P_m \cdot \sum_{l\in\mathbf{L}} M_l + P_e \cdot \sum_{l\in\mathbf{L}} E_l \tag{3.1}$$

subject to:

$$BWd_l \geq BW_l' \quad \forall l \in \mathbf{L}, \tag{3.2}$$

$$BWo \geq \sum_{l\in\mathbf{L}} BW_l' \tag{3.3}$$

$$E_l, M_l = 0, 1 \quad \forall l \in \mathbf{L}, \tag{3.4}$$

$$E_l + M_l \leq 1 \quad \forall l \in \mathbf{L}, \tag{3.5}$$

The optimization problem formalized above can be summarized as follows: our goal is to minimize the cost of the infrastructure. Such cost depends on the number of edge servers and on the number of multicast-enabled locations (Objective function 3.1). The goal is constrained by the conditions that, in each location, the bandwidth required by the clients must not exceed the bandwidth available for media download (constraint 3.2); furthermore, the outbound bandwidth on the origin server must be adequate, that is the origin server must be able to satisfy the clients that are not served by an edge server together with the bandwidth necessary to feed the edge server themselves (constraint 3.3). Additional constraints are related to the boolean nature of the control variables (constraint 3.4) and to the need to have either edge servers or multicast, but never both (constraint 3.5).

The problem is hard to solve due to the presence of a quadratic dependence on the decision variables in the definition of BW_l' (Equation 2). However, as in the contacts with the network provider emerged that $P_m \gg P_e$, we can simplify the problem by removing completely the support for multicasting. Basically we force $M_l = 0, \forall l \in \mathbf{L}$. This simplified problem can be solved in a straightforward way in two steps:

- we start by verifying the constraint 3.2 for every location; if the requirement is not met, we should place an edge server in that location;
- afterwards, we check constraint 3.3; if this second condition is not met, we introduce additional edge servers in the most bandwidth-consuming locations until the constraint is satisfied.

C. Access to On-demand Content

A different scenario considered in our analysis is that of access to on-demand content. Unlike the live-streaming scenario, in this case, we consider that each user can access a set of available stream at any time. In this case we aim to understand the probability of congestion arising, that is when we have too many downloads with respect to the available bandwidth. In case of congestion, the need for an edge server that can act as a cache for the media streams arises. For the sake of simplicity, in this model we do not explicitly model case where multiple encodings are available for each content, for a twofold reason. First, we assume that accessing on-demand content is performed mainly by devices that can consume any type of stream. Second, the model can still be applied to a case where multiple encodings are available by considering each stream with a different encoding as a separate content.

We defined the congestion as the case where the available bandwidth at location l is not sufficient to support the streaming. The maximum number of supported streams can be modeled as:

$$K_l = \left\lfloor \frac{BWd_l}{\overline{BW_t}} \right\rfloor \tag{4}$$

where BWd_l is the download bandwidth for location l and $\overline{BW_t} = \sum_{t\in\mathbf{T}} BW_t / |\mathbf{T}|$ is the average bandwidth for on-demand streams (if only high quality streams are available for on-demand fruition, we can simply consider the maximum bandwidth requirement).

Now, the probability of having congestion can be modeled as the probability of having more than K_l streams requested concurrently. If we define the probability of requesting a stream as p and we consider that the number of potential users at location l is $N_l = |\mathbf{C}_l|$ we have that:

$$P_l(cong) = \sum_{i=K_l+1}^{N_l} \binom{N_l}{i} p^i (1-p)^{N_l - 1} \tag{5}$$

where the Newton binomial coefficient is defined as:

$$\binom{N_l}{i} = \frac{N_l!}{i! \cdot (N_l - i)!} \tag{6}$$

This model has been used in preliminary analysis to understand the impact of the user behavior over the need of deploying edge servers in the P-CDN. However, these preliminary studies suggested that the live-streaming scenario is far more demanding than the on-demand one. For this reason, in the remaining of the paper we will focus just on supporting live streaming through the content delivery network.

D. Heuristic Solution

The previously-described models, even if able to capture exactly the main problem of CDN design, cannot be applied to its solution in a straightforward way. The main issues may be summarized as:

- A clear definition of the client requirements in terms of desired encoding is hard to obtain. Indeed, most clients can consume streams with different encodings, based on the actual available bandwidth. As a consequence, the most viable option is to ask for the highest possible quality level and accept a lower quality stream if bandwidth is not enough, rather than require a single encoding for each client.

- The number of clients may be very low for some company branches. Furthermore, the bandwidth may be rather limited as well. In this case, we would require an edge server to guarantee access to streaming media. However, the cost of installing such device may be not acceptable for these small branches. We will therefore consider a threshold to limit our study to just the medium/large branches.

To address the first problem, we propose a simple heuristic algorithm to assign the available bandwidth to the clients within the same location (Algorithm 1).

Algorithm 1 Client bandwidth allocation

```
Require: $\mathbf{C}_l$
Ensure: $t_c \ \ \forall c \in \mathbf{C}_l$
  for $t \in \texttt{sort}(\mathbf{T} \cup \{na\} \uparrow)$ do
    for $c \in \mathbf{C}_l$ do
      $BW'_l \leftarrow BW'_l - BW_{t_c} + BW_t$
      if $BW'_l \leq BWd_l$ then
        $t_c \leftarrow t$
      else
        break
      end if
    end for
  end for
```

Algorithm 1 starts with the lowest possible quality to allocate the bandwidth to the clients, and then increases the stream quality (and the bandwidth utilization) until we run out of networking resources. Let $\mathbf{C}_l$ be the set of clients in location l. The outer loop iterates over the possible stream encodings in ascending order of bandwidth consumption. While there is enough bandwidth, the algorithm assigns that bandwidth to the clients and updates the remaining bandwidth in the location. In our algorithm, we introduce the value na for the stream quality, meaning that the client cannot access any stream at any quality. Clearly, a stream quality na has a null bandwidth consumption.

When the algorithm stops, we may encounter one of the following four conditions:

- $BWd_l < \min(BW_t)$. This would mean that $t_c = na, \forall c$. In this case the considered location cannot support any stream, even at the lowest quality. No CDN benefit can be achieved here because, without a network upgrade no stream can be sent to the clients. The only possible solution in this case is to exclude the company branch from the analysis.
- $\exists c : t_c = na$. In this case not every client can access the stream even at the lowest quality. This is the case where an edge server is mandatory to avoid a situation where the consumption of live streams is possible only for a subset of the considered client population.
- $\exists c : t_c < \max(t_c)$. In this case every client can access to the stream. However, some of them (or all of them) can only consume a lower quality stream in order to avoid congestion on the location link. The presence of an edge server can fix this issue, but the need is less critical compared to the previous case.
- $t_c = \max(t_c) \forall c$. In this case every client can access to the stream at the highest possible quality.

To summarize, the edge server is useful or necessary only in the second and third case.

A second problem, as previously pointed out, is the presence of badly connected locations with a limited number of clients. In this case, even if it may be advisable to support the client access to the stream, the cost of an edge server may be too high. For this reason, we introduce a threshold parameter thr to control the number of clients in a given location and we define the following conditions to manage this trade-off:

- $|\mathbf{C}_l| \geq thr$: the location can be considered as a potential location for an edge server;
- $|\mathbf{C}_l| < thr$: the location is too small to host an edge server. For this reason, we force $E_l = 0$, preventing the possibility of placing an edge server in the location.

The parameter thr is used to introduce into the model the management policies concerning how much the company is willing to pay for supporting a high-quality delivery of content throughout the company branches. Preliminary meetings with the stakeholders (that is, the representatives of the company), suggested that, rather than trying to hard-code the company policies in our optimization model, an approach based on the ability to produce several *what-if* scenarios would have been more useful.

It is worth to note that the choice of whether a location should host or not an edge server is based purely on locally available information, such as the number of users and the available bandwidth. For this reason the optimization does not require complex algorithms, but the simple greedy heuristic proposed is sufficient to handle the problem.

E. Use of Satellite Locations

As a final part of the model, we take into account an evolution of the basic CDN model previously described. In particular, we introduce the idea that an edge server in a location may be used to feed a stream to clients in nearby locations without an edge server. It is worth to note that satellite locations are useful just to off-load the origin server from the need to serve locations that already do not need an edge server (that is when the client bandwidth allocation problem ends with the fourth outcome). To this aim, we introduce the notion of a *Satellite Location*, that is a location where the clients rely on the edge server of a different location for content delivery. In our model we consider satellite locations just as an additional feature, that can be either enabled or disabled based on the company policies without changing the CDN core layout.

To formalize the problem, let $e \in \mathbf{L_e}$ be a location hosting an edge server. We define $\mathbf{L}_{\mathbf{s}_e}$ as the set of satellite locations depending on the edge server in e. The edge server in e

will consume the upload bandwidth of the location to this aim. Hence, the main constraint for the definition of satellite locations is that the download bandwidth of the clients in the satellite locations must not exceed the upload bandwidth of the edge server; that is:

$$BWu_e > \sum_{s \in \mathbf{L_s}_e} BW'_s \tag{7}$$

Where we can approximate BW'_s with BWd_s that is an upper bound for BW'_s. The use of satellite locations impact the bandwidth consumption at the origin server (previously modeled as part of constraint 3.3). Indeed, the global bandwidth requirement for the origin server is:

$$BW'_o = \sum_{e \in \mathbf{L_e}} \max(BW_t) + \sum_{l \in \mathbf{L} - \mathbf{L_e}} BW'_l \tag{8}$$

where the first term is the bandwidth due to the edge servers and the remaining part is the bandwidth for locations without an edge server. The use of satellite locations reduces the demands on the origin server to:

$$BW'_o = \sum_{e \in \mathbf{L_e}} \max(BW_t) + \sum_{l \in \mathbf{L} - \mathbf{L_e} - \mathbf{L_s}} BW'_l \tag{9}$$

where $\mathbf{L_s}$ is the set of satellite locations.

The selection of satellite locations can be automated using the Algorithm 2 [7].

Algorithm 2 Management of Satellite locations [7]

```
Require: $\mathbf{L}$, $\mathbf{L_e}$, $niter$, $minBW$
Ensure: $\mathbf{L_s}_e, \forall e \in \mathbf{L_e}$
  $\mathbf{L_s} \leftarrow \mathbf{L} - \mathbf{L_e}$
  $\mathbf{L_s}_e \leftarrow \emptyset, \forall e \in \mathbf{L_e}$
  for $i \in [1, niter]$ do
    for $s \in \mathbf{L_s}$ do
      $e \leftarrow$ nearestEdge($s$, $\mathbf{L_e}$)
      Add $s$ to $\mathbf{L_s}'_e$
    end for
    for $e \in \mathbf{L_e}$ do
      for $s \in$sort($\mathbf{L_s}'_e$, from nearest) do
        if $BWu_e \geq BWd_s$ then
          Add $s$ to $\mathbf{L_s}_e$
          $BWu_e \leftarrow BWu_e - BWd_s$
          Remove $s$ from $\mathbf{L_s}$
        end if
      end for
      if $BWu_e < minBW$ then
        Remove $e$ from $\mathbf{L_e}$
      end if
    end for
  end for
```

In Algorithm 2, we refer to the set location where we have edge servers as $\mathbf{L_e}$; while $\mathbf{L}$ is the set of locations that may host clients. The location that do not host an edge server is the difference between the two sets, that is $\mathbf{L_s} = \mathbf{L} - \mathbf{L_e}$. The satellite locations may be selected from this list. Furthermore, we define $\mathbf{L_s}_e$ as the list of satellite location for each edge server e.

The first step of the algorithm is to divide the set of possible satellite locations $\mathbf{L_s}$ assigning each $s \in \mathbf{L_s}$ to the nearest edge server. We define the set $\mathbf{L_s}'_e$ as the list of *potential* satellite locations for that edge (first nested loop) in Algorithm 2.

Next, we iterate over the edge servers and we analyze the potential satellites from the nearest one. If the upload bandwidth is enough, we insert the potential satellite in the list of satellite locations for that edge. We update the available upload bandwidth in the edge server and we remove the newly inserted satellite location from the list of potential ones. If, at the end of the cycle, the bandwidth for the edge server is exhausted (that is, less than a given threshold $minBW$), we remove the edge server from the analysis, because it will not be able to support any more satellites.

The whole process is iterated $niter$ times to take into account the re-organization in the list of edge servers as their upload bandwidth is exhausted. In our preliminary experiments, we found that a limited number of iterations (typically $niter = 3$) are sufficient to guarantee a steady state where no additional satellite locations are found.

IV. Experimental Results

In this section we introduce the reference scenario that characterizes our case study. Through it, we aim to validate our proposal for a model that supports the design of P-CDN. Furthermore, we aim to analyze the impact of the main parameters of the model. We recall that the main output of our model concerns:

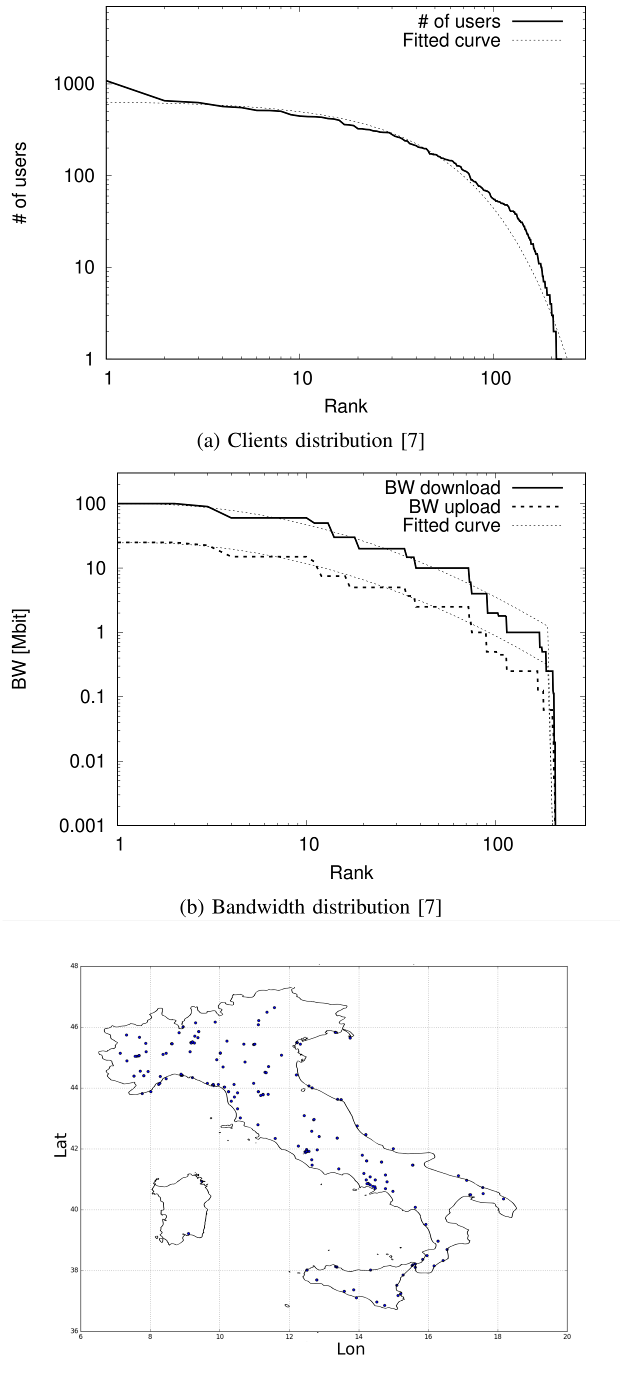
- placement of the edge server;
- definition of satellite locations.

In the following of the section, we introduce the reference scenario of our experiments, we provide a first set of results demonstrating the impact of a P-CDN infrastructure compared to a traditional origin-server based solution, and analyze how the threshold parameter thr affect these results.

A. Reference Scenario

The reference scenario is based on a real case provided by 47deck company; some of the most significant parameters are outlined in Figure 2. For privacy reason, we cannot provide results based on the original dataset supplied. To overcome this limitation, our analysis is based on a synthetic reconstruction of the initial network scenario, as in [7]. The new data set captures the characteristics of a network serving a company with nation-wide branches. We feed this scenario in our algorithms to study the placement of the edge servers composing the P-CDN. A further ex-post analysis is used to check that the case based on the synthetic (recreated) dataset provides results within 3% compared to the original ones.

Both the original and reconstructed datasets take into account 213 locations. For each location we record the number of users and the upload and download bandwidth. The clients in each location can be modeled as an exponential probability distribution. In Figure 2a [7] we show using a *log-log* scale both the number of users in each location (thick line) and the fitted curve (thin line). The fitting on the original data to obtain a probability distribution was carried out using the nonlinear



(a) Clients distribution [7]

(b) Bandwidth distribution [7]

(c) Company branches locations

Fig. 2. Reference scenario

TABLE II
IMPACT OF P-CDN

	P-CDN	no P-CDN
# of Locations		
Too small	99	0
Edge	77	0
Satellite	19	0
Origin	18	213
# of Clients		
LD	1577	1986
SD	94	94
HD	466	57

least-squares (NLLS) Marquardt-Levenberg algorithm. The fitted distribution is then used to re-create the dataset used in out experiments The bandwidths available for download and upload are shown in Figure 2b [7] using a solid and dashed line, respectively, together with the fitting curves. In this case, the function that best approximates the data is a truncated lognormal. Download and upload bandwidths show a similar behavior, with the upload bandwidth being typically 25% of the download one. As a significant note, our analysis suggests that the considered scenario is not characterized by the typical heavy-tailed distributions considered in other network descriptions [15] that may make the problem difficult to handle due to the high variability of the data. On the other hand, we can take advantage of truncated distributions, that have been proposed as a simplified description in other fields [16].

As part of the experimental setup, we show also the topology of the geographic locations considered in our dataset (Figure 2c): it is clear that the considered infrastructure spans over the whole country of Italy, with locations typically placed in the main cities. Referring to Figure 2c the X and Y axes report the Longitude (Lon) and Latitude (Lat) in the map, respectively.

Concerning the client requirements, we consider the following scenario with three types of encodings, each with a different bandwidth requirements. Specifically, we distinguish between:

- Low definition (LD), characterized by a bandwidth consumption of 512 Kb/s
- Standard definition (SD), characterized by a bandwidth consumption 1024 Kb/s
- High definition (HD), characterized by a bandwidth consumption of 1536 Kb/s

We consider that, for each location, only a fraction of the nominal bandwidth is available for streaming. Specifically, we assume that 60% of the upload/download bandwidth can be used for multimedia content delivery, while the remaining 40% is used by the normal operation carried out on the company network. It is worth to note that in our analysis, we explicitly discard (as pointed out in Section III-D) locations where the available is bandwidth below 512 Kb/s because these location would not be able to support any type of streaming even in the presence of an edge server.

B. Live Streaming Results

Based on the previously-described scenario, we now analyze the impact of the CDN in the considered setup. Using a simulation, we define a first design of the P-CDN infrastructure; using the design as a case study, we estimate the improvement of the user experience in terms of streaming video quality. We recall that a critical parameter of the model is the threshold value thr used to identify the locations that are too small in terms of users to host an edge server. In the results presented in this subsection we consider $thr = 50$, that provides a significant saving in terms of costs for the edge servers.

We present in Table II the number of edge servers, satellite location together with stream quality for the users with and without the P-CDN. This first set of experimental results was made available in [7] to show a first demonstration of the impact of a P-CDN architecture for the delivery of live media. The no P-CDN scenario is fairly simple as the origin server must cope with serving all the 213 locations considered in this study. As we introduce the P-CDN infrastructure we have 77 locations that are selected to host an edge server. These edge servers can further provide streaming support for 19 additional locations. Other 99 locations should host an edge server, but having just a few clients are discarded (we recall that in this analysis we consider $thr = 50$). The origin server is thus responsible for 18 locations together with the 99 small company branches previously mentioned.

As an additional contribution in our analysis, Figure 3 shows the placement of small locations, edge servers and satellite locations in a map (again the X and Y axis in the figure report, respectively, the longitude and latitude of every location). We observe the locations that are removed from our analysis due to their inadequate bandwidth or because there are too few clients in red and yellow color. Furthermore, we present the locations where we place an edge server using the purple color. In a similar way, we user a purple line to present the satellite locations, connecting the location itself with the edge server that feeds its clients.

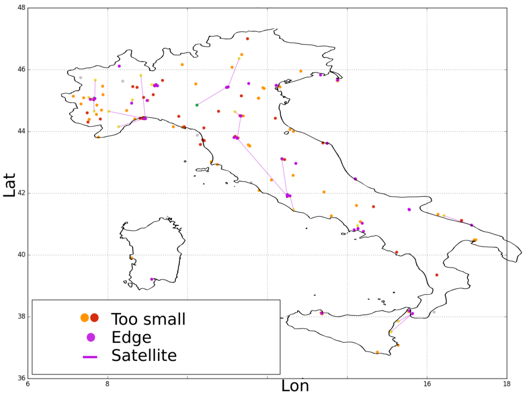
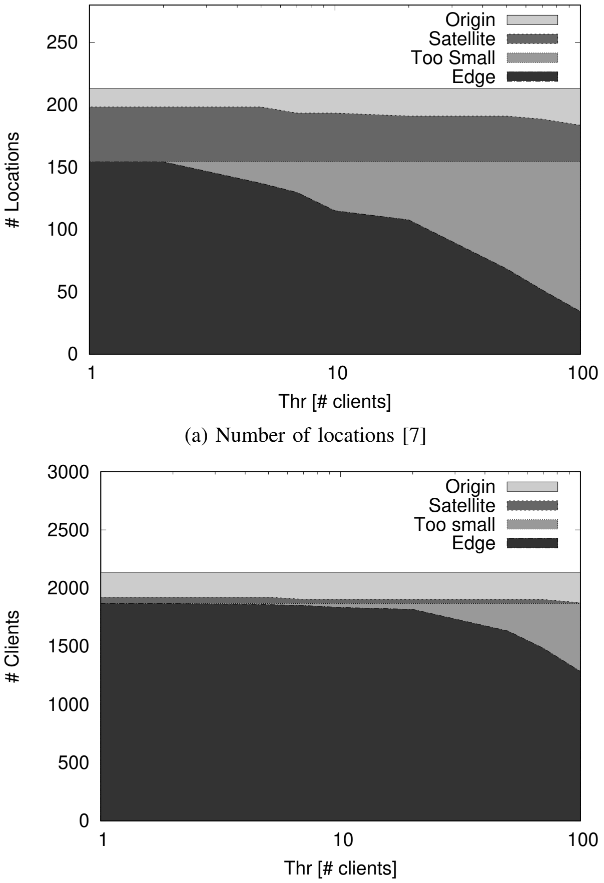


Fig. 3. CDN representation

If we consider the user experience, the adoption of the P-CDN has a clear impact. Without edge server, just a small fraction of clients can access the stream with something better than a low definition. Specifically only 94 (4%) and 57 (3%) clients can access to SD and HD video quality, respectively. If we consider the adoption of the P-CDN, we observe that the number of clients accessing the HD stream increases by more than eight times. The number of clients with a low quality feed remains high, but this is mainly due to the presence of a non-negligible number of clients that fall in the *Too small* company branches that are discarded as potential locations for edge servers.

C. Sensitivity to Threshold

We recall that the locations removed from the list of potential edge servers due to number of hosted clients have a major impact on the P-CDN layout, cost and performance. On an intuitive basis it is clear that more edge servers provide better quality on the streaming, but increase the cost of the infrastructure. The trade-off between cost and the quality is addressed using a threshold thr that identifies if a location is *too small* to justify the cost of an edge server. In order to provide a support for the decision-making process at the level of management, we consider useful to develop a series of *what-if* scenarios based on the different values of this parameter. The results of a detailed analysis on the impact of the thr parameter on the considered infrastructure are summarized in Figure 4.



(a) Number of locations [7]

(b) Number of clients [7]

Fig. 4. Locations and clients as a function of Thr

Figure 4a [7] provides a sensitivity analysis on the locations hosting edge servers of the P-CDN and on the corresponding satellite locations as a function of the thr parameter. Throughout this analysis we change our threshold thr to decide if a location is too small from 1 to 100 clients. The results are quite intuitive: as thr grows, the number of locations hosting an edge server is reduced and the corresponding number of locations considered *Too small* to host an edge server grows. The change in the number of edge servers has also an impact on the number of satellite locations: as the number of edge server is reduced, the number of satellite locations that can be supported decreases as well. Clearly, locations that do not host neither an edge server nor a satellite location are serviced directly by the origin servers.

In a similar way, Figure 4b [7] presents the same analysis applied to the number of clients. Specifically, we consider the number of clients corresponding to the previously defined four categories (that is, clients in locations *Too small*, clients serviced by edge servers, clients in satellite locations and clients serviced by the origin server). Again we run the tests for a threshold value $thr \in [1, 100]$. The main messages are comparable with the findings shown in Figure 4a. The main difference is the, considering the number of clients rather than the number of locations, the clients serviced by an edge server remains much larger as the number of locations with an edge server as thr grows. This effect can be explained considering that due to the nature of the threshold, locations in the *Too small* category host less clients than the locations with an edge server.

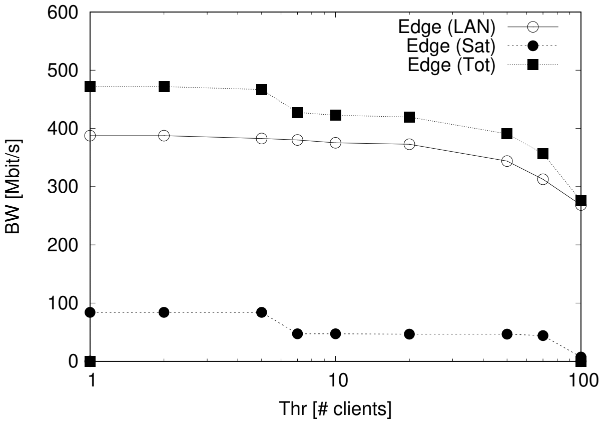


Fig. 5. Bandwidth utilization [7]

As a final contribution of our analysis, we present the bandwidth for edge servers throughout the P-CDN as a function of the threshold thr in Figure 5 [7]. We observe that, as the threshold grows, there is a reduction in the total bandwidth served by the edge servers due to their number reduction (line with square dots in Figure 5). To better understand this behavior, we show also a breakdown of the total bandwidth divided in bandwidth provided by the edge servers to support their local clients (line with white circles in Figure 5), and bandwidth used to feed the clients in the satellite locations (line with black circles). As expected, the total internal bandwidth of edge servers for servicing local clients (on the LAN) is decreased by the reduction in the number of locations hosting an edge server (as the bandwidth is a simple sum, reducing the number of edge servers decreases the number of client served). In a similar way, the drop in the number of satellite locations determines a reduction in the bandwidth devoted by the edge servers to clients outside their location due to the reduction of the satellite locations themselves (as pointed out when describing Figure 4a. This analysis is used in our industrial experience to define a tentative value for the threshold thr in the order of 50, that has been used in the preliminary results of our study.

V. Conclusions

This paper addresses the peculiar issue of designing a private CDN (P-CDN) for the delivery of streaming multimedia contents. Even if the main goal of designing a P-CDN is similar to that of a traditional CDN infrastructure, such scenario presents different characteristics: first, the underlying network infrastructure is outsourced to a third party that owns and operates it independently; second, content providers and content consumers belong to the same company. These characteristics may have a major impact on the expected workload and on the available choices for edge server placement, presenting peculiar challenges with respect to a traditional CDN scenario.

The proposal of the mathematical model to design the P-CDN coping with these problems represents the main contributions of this paper. Our solution has been validated through a simulator and the proposed model has been applied to a case study based on a real-world problem where a nation-wide company, consisting of branches located over two hundreds different cities, needs to deploy a P-CDN. Our experiments clearly show the existing trade-off between the effort to reduce the number of used edge servers and the need to place edge servers on locations with just a few clients.

Acknowledgements

The authors acknowledge:

- the support of 47Deck (`www.47deck.it`) for the definition of the reference scenario used in the model and in the experiments presented in the paper.
- the support of the University of Modena and Reggio Emilia through the project *S^2C: Secure, software-defined Cloud*.

References

[1] Akamai, "The world's largest and most trusted cloud delivery platform," 2017, https://www.akamai.com.

[2] Limelight, "Limelight networks," 2017, https://www.limelight.com.

[3] M. A. Kaafar, S. Berkovsky, and B. Donnet, "On the potential of recommendation technologies for efficient content delivery networks," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 74–77, Jul. 2013. [Online]. Available: https://doi.org/10.1145/2500098.2500109

[4] C. Canali, V. Cardellini, M. Colajanni, and R. Lancellotti, "Content delivery and management," in *Content Delivery Networks: Principles and Paradigms*, R. Buyya, M. Pathan, and A. Vakali, Eds. Springer, 2008. [Online]. Available: https://doi.org/10.1007/978-3-540-77887-5_4

[5] J. Sahoo, M. A. Salahuddin, R. Glitho, H. Elbiaze, and W. Ajib, "A survey on replica server placement algorithms for content delivery networks," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1002–1026, 2017. [Online]. Available: https://doi.org/10.1109/COMST.2016.2626384

[6] S. Spagna, M. Liebsch, R. Baldessari, S. Niccolini, S. Schmid, R. Garroppo, K. Ozawa, and J. Awano, "Design principles of an operator-owned highly distributed content delivery network," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 132–140, April 2013. [Online]. Available: https://doi.org/10.1109/MCOM.2013.6495772

[7] C. Canali, A. Corbelli, and R. Lancellotti, "Designing a private cdn with an off-sourced network infrastructure: model and case study," in *Proc. of Conference on Software, Telecommunications and Computer Networks (SOFTCOM)*, Supetar, Croatia, Sept. 2018. [Online]. Available: https://doi.org/10.23919/SOFTCOM.2018.8555766

[8] D. K. Krishnappa, M. Zink, and R. K. Sitaraman, "Optimizing the video transcoding workflow in content delivery networks," in *Proc. of ACM Multimedia Systems Conference*, ser. MMSys'15, Portland, Oregon, 2015. [Online]. Available: https://doi.org/10.1145/2713168.2713175

[9] C. Canali, V. Cardellini, and R. Lancellotti, "Content adaptation architectures based on squid proxy server," *World Wide Web Journal*, vol. 9, no. 1, pp. 63–92, 2006. [Online]. Available: https://doi.org/10.1007/s11280-005-4049-9

[10] Z. Li and G. Simon, "In a telco-cdn, pushing content makes sense," *IEEE Transactions on Network and Service Management*, vol. 10, no. 3, pp. 300–311, September 2013. [Online]. Available: https://doi.org/10.1109/TNSM.2013.043013.130474

[11] P. A. Frangoudis, L. Yala, A. Ksentini, and T. Taleb, "An architecture for on-demand service deployment over a telco cdn," in *IEEE International Conference on Communications (ICC)*, May 2016. [Online]. Available: https://doi.org/10.1109/ICC.2016.7510921

[12] C. Canali and R. Lancellotti, "Automated Clustering of Virtual Machines based on Correlation of Resource Usage," *Communications Software and Systems*, vol. 8, no. 4, pp. 102 – 110, Dec. 2012. [Online]. Available: https://doi.org/10.24138/jcomss.v8i4.164

[13] Akamai, "A content centric approach to delivering high quality mobile experiences," 2017, akamai White paper Library– https://content.akamai.com/PG2376-Mobile-Network-Solutions.html.

[14] M. Almashor, I. Khalil, Z. Tari, A. Y. Zomaya, and S. Sahni, "Enhancing availability in content delivery networks for mobile platforms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2247–2257, Aug 2015. [Online]. Available: https://doi.org/10.1109/TPDS.2013.2297927

[15] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An approach to universal topology generation," in *Proceedings of the Ninth International Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, ser. MASCOTS '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 346–. [Online]. Available: https://doi.org/10.1109/MASCOT.2001.948886

[16] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007, pp. 1–14. [Online]. Available: https://doi.org/10.1145/1298306.1298309

Claudia Canali received Laurea degree summa cum laude in computer engineering from the University of Modena and Reggio Emilia in 2002, and Ph.D. in Information Technologies from the University of Parma in 2006. She is a researcher at the Department of Engineering of the University of Modena and Reggio Emilia since 2008. Her research interests include cloud computing, social networks analysis, and wireless systems for mobile Web access. On these topics, Claudia Canali published more than fifty papers on international journals and conferences. She is a member of IEEE Computer Society. For additional information: http://weblab.ing.unimo.it/people/canali

Andrea Corbelli received the BsC in computer engineering in 2012 and the MsC in computer engineering in 2015 summa cum laude from the University of Modena and Reggio Emilia. He has been a research fellow and a teacher assistant at the University of Modena and Reggio Emilia from 2015 to 2017. Since then he has been working for other private companies, including Empirix, one of the world leading companies in providing end-to-end customer experience assurance software for mobile devices networks. His research interests include distributed systems and mobile network protocols."

Riccardo Lancellotti received the Laurea Degree in computer engineering summa cum laude from the University of Modena and Reggio Emilia in 2000 and the Ph.D. in computer engineering from the University of Roma "Tor Vergata" in 2003. He is a researcher at the University of Modena and Reggio Emilia since 2005. His research interests include geographically distributed systems, cloud computing, social networks and peer-to-peer systems. On these topics he published almost sixty papers on international journals and conferences. He is a member of the IEEE Computer Society and ACM. For additional information: http://weblab.ing.unimo.it/people/lancellotti