A Comparative Study of Content-Centric and Content-Distribution Networks: Performance and Bounds

Michele Mangili*†, Fabio Martignon*‡ and Antonio Capone†

* LRI, Université Paris-Sud
{michele.mangili, fabio.martignon}@lri.fr
‡ IUF, Institut Universitaire de France

† DEIB, Politecnico di Milano
capone@elet.polimi.it

***Abstract*—The Content-Centric Networking paradigm aims at improving the Quality of Service of the Internet by providing innovative features to better handle digital content distribution. A major step towards the success of this novel paradigm is to analyze and compare its performance with respect to the most popular ways in which content is disseminated in today's IP Internet. In this paper we give clear answers to this critical issue by proposing a methodology to assess how the innovative design of Content-Centric Networking behaves as opposed to the solution proposed by Content-Distribution Networks.**

We develop a novel optimization model to study the performance bounds of a Content-Centric Network, by addressing the joint *object placement* and *routing* problem. We further introduce three comparative models that well describe 1) a Content-Distribution Network, 2) a traditional IP-based network, and 3) a Content-Centric Network whose caches are prepopulated with given contents. To the best of our knowledge, our proposal is the first that studies the performance bounds of Content-Centric Networks by means of an optimization model.

Finally, we discuss the numerical results showing the performance bounds of this revolutionary paradigm. We discover that: 1) a Content-Centric Network with small caches can provide significant performance gains compared to a traditional IP-based network; 2) for large amounts of caching storage, the benefits of using sophisticated *cache replacement policies* are dramatically reduced and 3) in some scenarios, a Content-Distribution Network with few replica servers can perform better than a Content-Centric Network, even when the total amount of available caching storage is exactly the same.

I. Introduction

The amount of digital contents that users access on an everyday basis is raising exponentially, as envisioned by Cisco forecasts [1]. Furthermore, we are assisting to a radical change in the way people use the network: the Internet has become much more than just a communication infrastructure, up to the point that some authors have defined the global network as a "*platform for business and society*" [2].

This change poses new requirements on the Internet itself, which was not specifically designed to perform content distribution [3]. In order to tackle this issue, Content-Distribution Networks (CDNs), such as Akamai, have become a vital layer in the architecture of any content provider as they make it possible to distribute content in today's IP Internet in an efficient way. The driving principle of CDNs is that content requests are not directly served by the *origin* server owned by the content provider, but they are instead mediated by the CDN infrastructure. The CDN operator owns a given number of *surrogate servers*, scattered all over the world, which thus perform content caching and replication, improving the Quality of Service (QoS) of the consumers by serving the requests of the clients in the neighborhood [4]. One of the main features of CDNs is that they do not change the current key network protocols, but they rather offer countermeasures to address the peculiar characteristics of the Internet infrastructure that limit its effectiveness when performing content distribution.

An orthogonal approach is the one recently proposed by the research community on Content-Centric Networking (CCN) [5]. CCN is a novel design for the Next Generation Network that aims at overcoming the current limitations of the Internet, by providing new protocols centered around the data itself. The CCN paradigm proposes to change the addresses of the packets which should point directly to the *data* that has to be retrieved rather than the *location* where such data is stored. Among the advantages obtained by changing the addressing space, the performance gain stems out to be the most relevant achievement. In addition to that, further benefits, such as improved security [6] and better mobility support [7], will be provided as default features of the network.

In order to foster the diffusion of the Next-Generation Content-Centric Networks, it becomes vital to understand their performance bounds, especially when compared with the current CDN strategy. To this end, in this paper we provide a theoretical framework to study and compare the performance of this novel class of networks with respect to the strategies that are used nowadays to distribute content in the Internet. The contribution of this paper is threefold:

1) We formulate a novel optimization model to study the performance bounds of a Content-Centric Network, solving the joint *object placement* and *routing* problem.
2) We propose a comparative model to represent a similar scenario in a Content-Distribution Network, by addressing the joint *server* and *object placement* with *routing* problem.
3) We gauge the performance improvements achieved by a Content-Distribution Network with respect to a Content-Centric Network and we show that when the total amount of caching storage is the same, CDN can have slightly better performance than CCN.

Extensive numerical analysis and comparison between the different models complement the theoretical framework.

This paper is structured as follows: in Sec. II we discuss related works. Sec. III contains a background on the features of Content-Centric Networks relevant to our scenario. In Sec. IV we formulate the optimization model for Content-Centric Networks and illustrate the proposed comparative extensions. Numerical results are discussed in Sec. V. Finally, concluding remarks are presented in Sec. VI.

II. Related Work

This section summarizes the most notable works related to our proposition; in particular, Sec. II-A and Sec. II-B survey the literature on performance evaluation for Content-Distribution and Content-Centric Networks, respectively.

A. Performance Evaluation for Content-Distribution Networks

Optimization models have extensively been used to study the performance of Content-Distribution Networks, in particular they are extremely popular to address the *object placement* problem, also known under the name of *replica placement* problem [8], [9], [10], [11], [12], [13].

In particular, in [8], Baev et al. provide a 10-approximation algorithm for the *object placement* problem, with the precise aim to minimize the total cost given by both the access and the storage costs. Their model determines not only the optimal data replication in the caches, but also chooses the best client allocation to them. In [9], Kangasharju et al. provide four heuristics to solve the *object replication* problem by minimizing the average number of Autonomous Systems traversed in order to serve the requests. Linear optimization techniques are used also in [10] to study the performance of a Content-Distribution Network modeled as a hierarchical cache system with a single origin server.

Our work differs from [8], [9] and [10] in four main characteristics: 1) we do not explicitly take into account nodes cooperation, but our model finds the optimal solution independently of the fact that nodes might cooperate or not; 2) we model a scenario where many origin servers are available, each of which stores a different subset of the objects, thus a client might be served by many *replica* servers; 3) we believe that the tree topology used in such works is an oversimplification since the consumers might generate requests for many objects published by different servers, therefore we model the network as an undirected graph; 4) our model jointly solves the *object allocation* as well as the *routing* problem.

B. Performance Evaluation for Content-Centric Networks

One of the design goals of Content-Centric Networks is to improve the Quality of Service of the Internet, by providing better support for content distribution [5]. It is therefore necessary to assess whether this objective can be fulfilled using the architectures that have been proposed so far. This evaluation was conducted in [14], [15], [16].

Rossi and Rossini evaluated in [14] the performance of CCN by means of simulating the behavior of such a network under the realistic assumption that the object catalog is composed by videos published on YouTube. They built the *CcnSim* simulator and performed extensive analysis with it, varying the network topology as well as the most relevant parameters that characterize the traffic demands and the caches. In [15], Jacobson et al. provide a performance comparison of a Content-Centric Network with respect to an IP-based solution when running a VoIP application. They show that both the IP-based network and the Content-Centric solution have similar performance. The first performance evaluation of CCN on a real testbed has been done by Crowley et al. using the Open Network Laboratory (ONL) and results were presented in [16]. The authors performed the comparison of CCN with respect to a classical HTTP proxy for the download of a file, and showed that with the CCNx prototype available in 2011, CCN was 10 times slower than HTTP, since performing lookup of chunk names increases the computational overhead of packet forwarding, as described in [17].

The present article differs from the previous approaches because we study the behavior of CCN using neither a simulator nor a testbed but rather a performance model. This choice gives us the chance to understand the theoretical performance bounds of this novel class of networks without having to deal with the implementation issues that might arise as shown in [16]. Moreover, our focus is to provide a comparison between the behavior of a Content-Centric and a Content-Distribution Network, under heavy traffic conditions.

III. Content-Centric Networks

This section introduces the features provided by Content-Centric Networks (CCN) that are relevant for our proposal. A comprehensive description of CCN can be found in [5].

The communication model proposed by CCN is characterized by the fact that instead of having the host addresses of the sender and the receiver, the packets contain only the *content name* field, which represents the identifier of the data that was requested. CCN has two distinguished packet types: 1) Interest and 2) Data packets. The former does not contain the actual data, but it declares that a node is willing to access a given object; the latter associates to a *content name* the corresponding bits of data.

The structure of a CCN router is characterized by three tables: 1) the Pending Interests Table (PIT); 2) the Content Store (CS) and 3) the Forwarding Interest Table (FIB).

The PIT is the data structure responsible for memorizing the list of Interests previously forwarded, but not yet answered. Interests might arrive from physical hardware interfaces as well as logical applications running on the node itself. For this reason, in CCN the generalization of an interface is called "*face*": a logical interface. The PIT stores the *faces* from which Interests were originally received. This is done in order to implement *reverse path forwarding*: as soon as a router receives a Data packet, it checks the PIT and forwards the packet on the same faces from which Interests for that object arrived. The CS is the data structure responsible for caching Data packets. It is used to implement universal in-network caching. When an Interest arrives, the router will initially query the CS; in the case of a cache hit, the router will be able to directly serve the data. The FIB comes into play when cache miss happens: it contains the next-hop information for prefix names. In this context, a *Consumer* is a node that requests some content to the network, while a *Producer* is a node that can reply to an Interest by directly providing the associated Data packet.

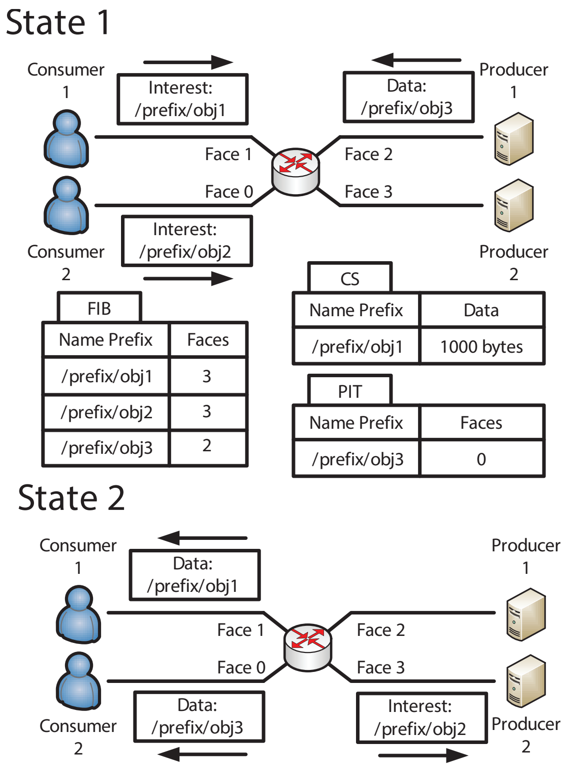
An example showing the behavior of a CCN router is depicted in Fig. 1. In *State 1*, the router receives two Interests and one Data packet. As shown in Fig. 1, the Interest for object `/prefix/obj1` is directly served by the router since it is available in its *Content Store* (CS). The Interest for `/prefix/obj2` will be forwarded to *Face 3* since it is the destination available in the *Forwarding Information Base* (FIB). Lastly, the Data packet for `/prefix/obj3` will be forwarded to *Face 0*, as written in the *Pending Interests Table* (PIT). In *State 2*, the transition to the next state is represented: as described, the router forwards one Interest and two Data packets.

IV. Optimization Models

This section describes the optimization model we propose to jointly determine the optimal traffic routing and cache management (i.e., the object placement in the nodes' cache) in CCN. We then introduce three comparative models used to study: 1) the performance of a Content-Distribution Network (CDN); 2) an IP-based network and 3) a CCN whose caches are pre-populated with given objects.

A. Joint Routing and Cache Management in Content-Centric Networks

The goal of our proposed optimization model is to jointly optimize the traffic routing and the content (object) placement in the routers' cache, in order to minimize the overall bandwidth occupied on all network links. This permits to study the benefit that a CCN architecture can offer, in terms of bandwidth saving, with respect to the current IP Internet and CDN infrastructures. Therefore, the output of the model encompasses: 1) the set of objects cached in every node; 2) the amount of traffic traversing every link of the network.



FIB

Name Prefix	Faces
/prefix/obj1	3
/prefix/obj2	3
/prefix/obj3	2

CS

Name Prefix	Data
/prefix/obj1	1000 bytes

PIT

Name Prefix	Faces
/prefix/obj3	0

Figure 1. Example illustrating the behavior of a CCN router. Two Interests and one Data packet are received by the router in *State 1*. Given the information contained in the *Pending Interests Table* (PIT), the *Content Store* (CS) and the *Forwarding Information Base* (FIB), in *State 2* the router forwards two Data packets and one Interest.

The CCN is modeled as an undirected graph $G(V, E)$, where V is the set of vertexes (nodes), while E is the set of edges (links). A vertex can either act as a consumer, a producer or a router. Let $\mathcal{C}$ be the set of consumers, $\mathcal{P}$ the set of producers and $\mathcal{R}$ the set of CCN routers. We assume that each CCN router performs not only packet forwarding, but also caching. Thus we have that $V = \mathcal{C} \cup \mathcal{P} \cup \mathcal{R}$.

We further assume that each consumer $c \in \mathcal{C}$, as well as each producer $p \in \mathcal{P}$, is connected to a single router $r \in \mathcal{R}$, whereas each router may have many links to other routers. We denote with b_{rc}, where $r \in \mathcal{R}$ and $c \in \mathcal{C}$, the link capacity of router-consumer links. Similarly, the producer-router link capacity is denoted by b_{pr}, with $p \in \mathcal{P}$ and $r \in \mathcal{R}$, and the router-router link capacity is given by $b_{r_1 r_2}$, where $(r_1, r_2) \in \mathcal{R} \times \mathcal{R}$.

The first class citizens in CCN are the objects; in line with the literature [14], we assume that objects have the same size s. We denote with $\mathcal{O}$ the set of objects, where the cardinality $|\mathcal{O}|$ can be in the order of 10^8 (or even more), as mentioned in [18]. Since working with such a huge number of objects is practically unfeasible in an ILP model, we make the realistic assumption that objects can be grouped in two sets: 1) $\mathcal{O}^c$ is the set of *most popular* (or "cachable") objects, while 2) $\mathcal{O}^u$ is the set of *least popular* (or "non-cachable") objects. These sets are such that $\mathcal{O} = \mathcal{O}^c \cup \mathcal{O}^u$. This hypothesis well represents the fact that, due to the limited cache sizes and the content requests popularity of every object, only the subset of the most popular objects will likely be stored in the distributed caches. The demand of every consumer for every object is known, and is denoted by d_{co}, where $c \in \mathcal{C}$ and $o \in \mathcal{O}$.

The binary variable $a_{po} \in \{0, 1\}$ represents the availability of an object $o \in \mathcal{O}$ at a given producer $p \in \mathcal{P}$, and is such that:

$$a_{po} = \begin{cases} 1, & \text{if object } o \text{ is available at producer } p \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The optimization model will compute the optimal object allocation. Formally, we describe this allocation using the binary variable $x_r^o \in \{0, 1\}$, defined for all pairs $(r, o) \in \mathcal{R} \times \mathcal{O}^c$. In particular, it is such that:

$$x_r^o = \begin{cases} 1, & \text{if router } r \text{ stores cachable object } o \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The model also computes the optimal traffic routing, and hence the flow for every link. The router-to-router flow for data sent from r_1 to r_2 will be denoted by $y_{r_1 r_2}^o$ for each $(r_1, r_2, o) \in \mathcal{R} \times \mathcal{R} \times \mathcal{O}$. The producer-to-router and the router-to-consumer flows will instead be denoted respectively by y_{pr}^o for each $(p, r, o) \in \mathcal{P} \times \mathcal{R} \times \mathcal{O}$ and by y_{rc}^o for each $(r, c, o) \in \mathcal{R} \times \mathcal{C} \times \mathcal{O}$. Table I summarizes the notation used in this paper.

Given the above definitions and assumptions, we formulate the optimal Object Allocation and Routing model (OAR) in CCN as follows:

$$\min \sum_{\forall o \in \mathcal{O}} \left(\sum_{\substack{\forall r_1 \in \mathcal{R} \\ \forall r_2 \in \mathcal{R}}} y^o_{r_1 r_2} + \sum_{\substack{\forall p \in \mathcal{P} \\ \forall r \in \mathcal{R}}} y^o_{pr} + \sum_{\substack{\forall r \in \mathcal{R} \\ \forall c \in \mathcal{C}}} y^o_{rc} \right) \tag{3}$$

subject to:

$$\sum_{c \in \mathcal{C}} y^o_{r_1 c} + \sum_{r_2 \in \mathcal{R}} y^o_{r_1 r_2} \leq Q \cdot x^o_{r_1} + \sum_{r_2 \in \mathcal{R}} y^o_{r_2 r_1} + \sum_{p \in \mathcal{P}} y^o_{p r_1} \qquad \forall (r_1, o) \in \mathcal{R} \times \mathcal{O}^c \tag{4}$$

$$\sum_{c \in \mathcal{C}} y^o_{r_1 c} + \sum_{r_2 \in \mathcal{R}} y^o_{r_1 r_2} \leq \sum_{r_2 \in \mathcal{R}} y^o_{r_2 r_1} + \sum_{p \in \mathcal{P}} y^o_{p r_1} \qquad \forall (r_1, o) \in \mathcal{R} \times \mathcal{O}^u \tag{5}$$

$$\sum_{\forall o \in \mathcal{O}} y^o_{r_1 r_2} \leq b_{r_1 r_2} \qquad \forall (r_1 r_2) \in \mathcal{R} \times \mathcal{R} \tag{6}$$

$$\sum_{\forall o \in \mathcal{O}} y^o_{pr} \leq b_{pr} \qquad \forall (p, r) \in \mathcal{P} \times \mathcal{R} \tag{7}$$

$$\sum_{\forall o \in \mathcal{O}} y^o_{rc} \leq b_{rc} \qquad \forall (r, c) \in \mathcal{R} \times \mathcal{C} \tag{8}$$

$$\sum_{\forall r \in \mathcal{R}} y^o_{rc} = d_{co} \qquad \forall (c, o) \in \mathcal{C} \times \mathcal{O} \tag{9}$$

$$\sum_{\forall o \in \mathcal{O}^c} x^o_r \leq S \qquad \forall r \in \mathcal{R} \tag{10}$$

$$y^o_{pr} \leq b_{pr} \cdot a_{po} \qquad \forall (p, r, o) \in \mathcal{P} \times \mathcal{R} \times \mathcal{O}^c \tag{11}$$

$$x^o_r \in \{0, 1\} \qquad \forall (r, o) \in \mathcal{R} \times \mathcal{O}^c \tag{12}$$

$$y^o_{r_1 r_2} \geq 0 \qquad \forall (r_1, r_2, o) \in \mathcal{R} \times \mathcal{R} \times \mathcal{O} \tag{13}$$

$$y^o_{pr} \geq 0 \qquad \forall (p, r, o) \in \mathcal{P} \times \mathcal{R} \times \mathcal{O} \tag{14}$$

$$y^o_{rc} \geq 0 \qquad \forall (r, c, o) \in \mathcal{R} \times \mathcal{C} \times \mathcal{O}. \tag{15}$$

The objective function (3) minimizes the total traffic transferred over all network links. In (4) we set the flow balance constraints for *cachable* (popular) content. At every router in the network and for every cachable object, the sum of the incoming data and the bandwidth used by the node to serve cached content must be equal to the total outgoing data. To express this condition, we make use of the Q parameter, which is a large constant value. We then express a similar condition for *non-cachable* (least popular) content, in the set of constraints (5). Constraints (6), (7) and (8) check that the traffic exchanged between two nodes does not exceed the available link capacity. The set of constraints (9) makes sure that the network serves the consumers' demands. Each router in the network can store up to S objects, as enforced by constraints (10). Constraints (11) make sure that a producer is serving only the content that it is publishing in the network. In other words, a provider can only serve the content it owns. Finally, we add to the model the binary constraints (12) on x^o_r, as well as the positivity constraints (13)-(15) on the flow variables.

In the following sub-sections, we discuss the extensions to OAR, to deal with a scenario where: 1) the distributed CCN cache is replaced by a limited number of CDN servers, 2) the routers do not have any cache (which well represents today's IP Internet) and 3) the CCN routers' cache is populated randomly.

Table I. SUMMARY OF THE NOTATION USED IN THIS PAPER.

Parameters of the ILP Model	
$\mathcal{C}$	Set of Consumers
$\mathcal{P}$	Set of Producers
$\mathcal{R}$	Set of Routers
$\mathcal{O}$	Set of Objects
$\mathcal{O}^c$	Set of most popular ("cachable") objects
$\mathcal{O}^u$	Set of least popular ("non-cachable") objects
b_{rc}	Link capacity between router r and consumer c
b_{pr}	Link capacity between producer p and router r
$b_{r_1 r_2}$	Link capacity between router r_1 and router r_2
S	Maximum number of objects that each router can cache
d_{co}	Traffic demand generated by consumer c for object o
a_{po}	0-1 Parameter that indicates whether producer p is publishing object o
Q	A large number
N	Maximum number of CDN nodes that can be deployed
m	The maximum memory that a CDN node can use for caching
M	The total memory shared by all CDN nodes for caching

Decision Variables of the ILP Model	
x^o_r	0-1 Variable that indicates if router r is caching object $o \in \mathcal{O}^c$
$y^o_{r_1 r_2}$	Data flow related to object o from router r_1 to the neighbor router r_2
y^o_{pr}	Data flow related to object o from producer p to router r
y^o_{rc}	Data flow related to object o from router r to consumer c
l_r	0-1 Variable that indicates if a CDN node is installed in router r
z^o_r	0-1 Variable that indicates if the CDN node at router r caches object o

B. Modeling a Content-Distribution Network

For our performance model, one of the most relevant differences between a Content-Centric and a Content-Distribution Network is that, in the former, ideally any router in the network behaves as a cache, while in the latter, dedicated content servers are distributed in the network. Due to scalability constraints, it is reasonable to assume that the total amount of memory that will be deployed in each CCN router is limited. This requirement is related to the fact that a CCN router has to work at wire-speed and the larger the cache size is, the slower the memories are, thus imposing a constraint on the maximum size of the memories that can be used. This problem is mitigated instead in a CDN server, which has to respond to every request introducing the shortest delay possible, but can work at much slower rates when compared to a core router. This characteristic makes it possible to work with larger memories.

The model for a CDN is an extension of the OAR model illustrated above, where a given number of CDN nodes (i.e., caches) can be deployed in the network. Let N be the maximum number of CDN nodes; we denote with m the maximum quantity of memory that can be installed in a given CDN node, while M is the total memory that will be shared by all of them.

Two additional binary decision variables are added in this case with respect to OAR. The first is denoted by $l_r \in \{0, 1\}$, $\forall r \in \mathcal{R}$, and discriminates the routers at which a CDN node is connected:

$$l_r = \begin{cases} 1, & \text{if a CDN node is connected to router } r \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The second variable represents the objects cached by a CDN node connected to a given router, and is denoted by z^o_r, $\forall (r, o) \in \mathcal{R} \times \mathcal{O}^c$:

$$z^o_r = \begin{cases} 1, & \text{if CDN node at router } r \text{ stores cachable object } o \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

Thus, we can formulate the Optimal CDN Design model (OCDN) as follows:

$$\min \sum_{\forall o\in\mathcal{O}} \left(\sum_{\substack{\forall r_1\in\mathcal{R}\\ \forall r_2\in\mathcal{R}}} y^o_{r_1r_2} + \sum_{\substack{\forall p\in\mathcal{P}\\ \forall r\in\mathcal{R}}} y^o_{pr} + \sum_{\substack{\forall r\in\mathcal{R}\\ \forall c\in\mathcal{C}}} y^o_{rc} \right) \tag{18}$$

subject to constraints (5)-(9), (11), (13)-(15) and

$$\sum_{c\in\mathcal{C}} y^o_{r_1c} + \sum_{r_2\in\mathcal{R}} y^o_{r_1r_2} \le Q\cdot z^o_{r_1} + \sum_{r_2\in\mathcal{R}} y^o_{r_2r_1} + \sum_{p\in\mathcal{P}} y^o_{pr_1} \quad \forall(r_1,o)\in\mathcal{R}\times\mathcal{O}^c \tag{19}$$

$$\sum_{\forall o\in\mathcal{O}^c} z^o_r \le m \qquad \forall r\in\mathcal{R} \tag{20}$$

$$\sum_{\substack{\forall r\in\mathcal{R}\\ \forall o\in\mathcal{O}^c}} z^o_r \le M \tag{21}$$

$$\sum_{\forall r\in\mathcal{R}} l_r \le N \tag{22}$$

$$\sum_{\forall o\in\mathcal{O}^c} z^o_r \le Q\cdot l_r \qquad \forall r\in\mathcal{R} \tag{23}$$

$$z^o_r \in \{0,1\} \qquad \forall(r,o)\in\mathcal{R}\times\mathcal{O}^c. \tag{24}$$

The objective function (18) as well as constraints (5)-(9), (11), (13)-(15) are shared by both OAR and OCDN. By having the same objective function, we make sure that a fair comparison will be performed in the two scenarios, since the metric does not change. The other common constraints represent basic properties that the network has to guarantee in any case (i.e., the flow balance for objects not cached at every node; bandwidth limits on the links; producers offering only the objects that they store; positivity constraints).

In (19) we express the flow balance condition for objects that will be cached in CDN servers: a CDN node can directly serve the objects it stores. Constraints (20) make sure that each CDN node does not store more objects than the maximum quantity allowed. At the same time, the total cache memory shared by the CDN should not exceed the maximum value M, as enforced by constraint (21). In constraint (22) we make sure that the number of CDN nodes does not exceed the limit N, while in constraints (23) we make sure that the binary variable l_r is set to 1 if a CDN node is connected to router r. Finally, we add to the model the binary constraints (24).

C. IP Model and Random Cache Content-Centric Model

Hereafter, we illustrate the changes to the OAR optimization model in order to describe: 1) the behavior of a network that does not provide in-network caching like in today's IP Internet, and 2) a Content-Centric Network whose caches have been pre-initialized.

In the first case, each node in the network does not have caching capabilities anymore but it performs content forwarding towards the closest destination. By removing constraints (4), (10) and (12), we obtain an optimization model designed for a network that does not implement in-network caching; the problem is therefore reduced to the simpler scenario of a *routing* problem. As a positive side effect of this change, the model becomes a Linear Program. It is important to note that the model still supports multicast routing which will be leveraged when a link is saturated. This model thus provides a lower bound on the total bandwidth required to distribute the content in an IP-based network such as the Internet.

We propose another extension to study the scenario where the CCN caches have been pre-initialized with random content. In this case the value of the bi-dimensional matrix x^o_r is given as an input of the problem. This change well represents a scenario where the distributed cache is available but it is not possible to optimally allocate the objects to the nodes.

V. Numerical Results

In this section, numerical results are extensively analyzed and discussed for the different scenarios considered in this paper. In particular, we first describe (Sec. V-A) the methodology used to evaluate our models, while in Sec. V-B we discuss the numerical results.

A. Methodology

We performed an extensive evaluation of the models we designed, using the topologies also considered by Rossi et al. in [14]:

1) The Abilene network (11 routers, 14 links);
2) The GÉANT network (37 routers, 56 links);
3) A Random-Geometric Graph (26 routers, 60 links).

A graphical representation of the Abilene topology is provided in Fig. 2. Unless otherwise specified, in the following we will refer to this network, since similar trends have been obtained with the other topologies.

We fix the number of producers and consumers to 10 and 25, respectively. Each producer/consumer is connected to one router uniformly selected between those available in the network. We further set the bandwidth limits on all the links to 10 Gbit/s. The total traffic demand is the same for all the consumers. We analyze five possible demands in the set $\{1,2,3,4\}$ Gbit/s. We model the content popularity of every object by means of a Zipf distribution, since it is one of the most popular models proposed in the literature for

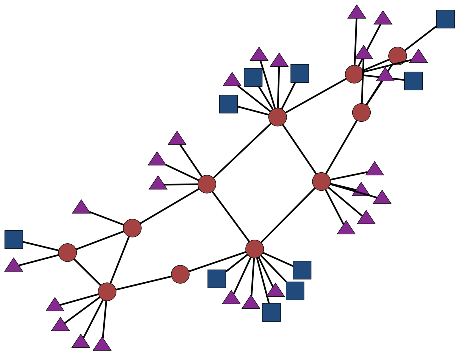
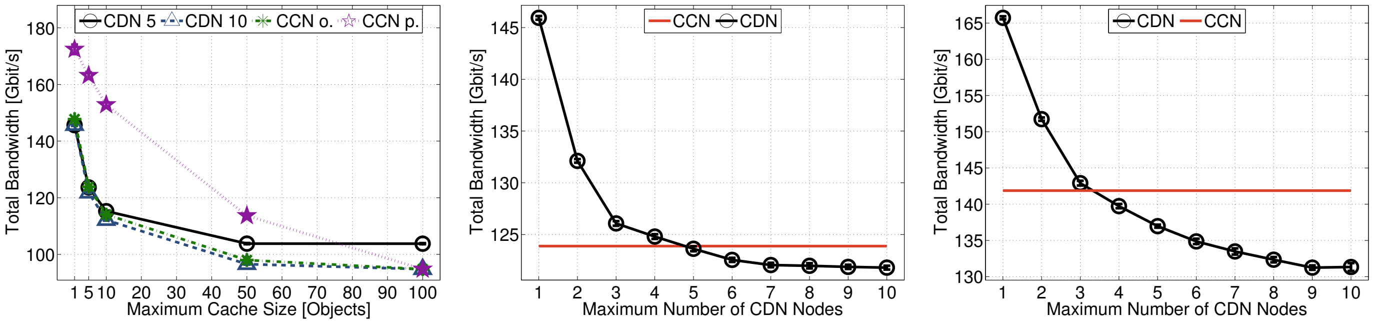


Figure 2. Abilene topology used for the numerical results. Purple triangles represent consumers; red circles represent routers; blue squares represent producers.

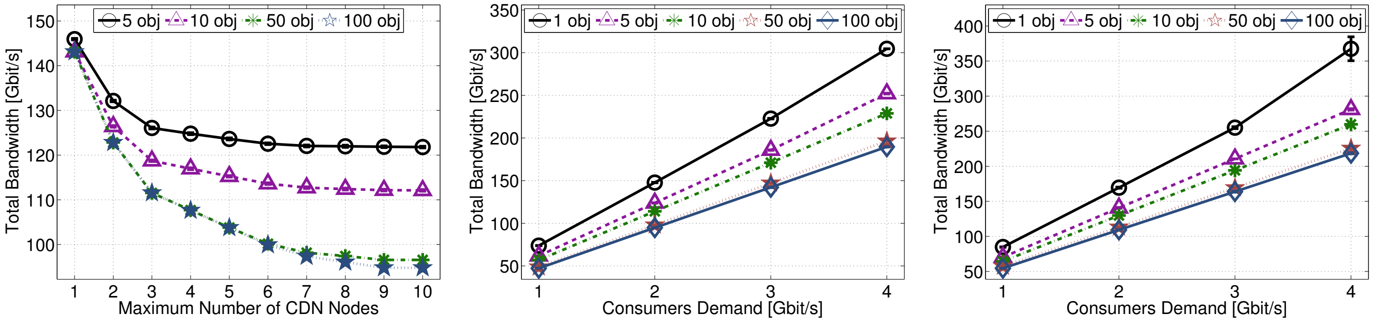


(a) Cache Size Comparison - Topology: Abilene

(b) CCN vs. CDN - Topology: Abilene

(c) CCN vs. CDN - Topology: GÉANT

Figure 3. Fig. 3(a) represents the total used bandwidth as a function of the cache size for a CDN network with 5 and 10 nodes, a CCN optimal network (CCN o.) and a CCN Network with randomly pre-populated caches (CCN p.). Fig. 3(b) and 3(c) show a comparison between CCN and CDN where the same total caching storage is allocated.



(a) CDN Nodes Comparison - Topology: Abilene

(b) CCN - Topology: Abilene

(c) CCN - Topology: Random

Figure 4. Fig. 4(a) represents the behavior of a CDN as a function of the number of nodes and the amount of available caching storage. Fig. 4(b) and 4(c) show the behavior of a CCN network as a function of the demands, for the Abilene and the Random Geometric Graph topology, respectively.

this type of analysis. A common value for the alpha exponent of such distribution is 1.2, as shown in [18], where a formal description of some of the basic properties of this distribution can also be found. Since we use a realistic cardinality of objects (equal to 10^8), we group them in *popularity classes*. We thus discriminate the objects according to their popularity in two groups: 1) the most popular objects and 2) the least popular objects, as illustrated in Sec. IV. Since we assume that the average size of an object is 100 Mbytes (as shown in [18]) and a reasonable size for a CCN cache is 10 Gbyte, the number of the most popular objects is set to 100.

Content providers are eager to protect their delivered contents; for this reason, we assume that each object belonging to the "*most popular*" class can be served only from one single producer. On the other hand, to deal with the huge cardinality of the least popular objects, we group them in classes and we assign each class to one producer by balancing the load on all of them. To do that, we map the least popular object $o \in \mathcal{O}^u$ to the producer $p \in \mathcal{P}$ if and only if $p = o \bmod |\mathcal{P}|$.

The performance metric that we study in this section is the *total bandwidth* consumed in the network, as defined in the objective function of our optimization models, (3) and (18). For each analysis we generated 10 different demand profiles and we computed the mean total bandwidth; we also obtained very narrow 99% confidence intervals, shown in the figures.

B. Result Analysis

Using the network model without caches, as described in Sec. IV-C, we discovered that all the topologies support only a demand up to 1 Gbit/s per consumer. On the other hand, when we introduce even very small caches (i.e., 1 object per router), we can easily accommodate demands reaching 4 Gbit/s per consumer. The presence of the cache can thus significantly improve the network performance by accommodating much higher demands even when just a modest amount of memory is available.

Fig. 3(a) illustrates a comparison between 1) the behavior of a CDN network with 5 and 10 nodes, 2) the optimal CCN model (CCN o.) and 3) a scenario where the CCN is randomly pre-populated (CCN p.). In all these models, passing from the case where 1 object can be cached at every node up to 10 objects makes the objective function decrease very sharply. Even if we further increase the cache size, the objective function saturates and minor benefits are experienced when passing to 50 or 100 objects cached per node. This trend gives insights regarding the fact that it is not worth to allocate very large amounts of memory to the caches due to a modest performance gain. This behavior can be explained by the intrinsic characteristics of the Zipf distribution: there exist many objects that are seldom accessed. Furthermore, in Fig. 3(a) the gap between the optimal CCN allocation (CCN o.) and the random objects allocation (CCN p.) is reduced as the *maximum cache size* increases. It is thus evident that the benefits of having sophisticated *cache replacement policies* depend on the number of objects that can be cached: the more caching storage is available, the more limited the performance gain will be.

Fig. 3(b) and 3(c) show a direct comparison between CDN and CCN (the horizontal line) for the Abilene and the GÉANT topology, respectively. The demands are set to 2 Gbit/s, while the total storage available is the same for both the topologies and equal to 5 objects per router. In Abilene (Fig. 3(b)), 5 CDN nodes are sufficient, on average, to make CDN be equivalent to CCN in terms of performance, whereas the same limit is decreased to a value between 3 and 4 nodes in the GÉANT topology (Fig. 3(c)). If the number of CDN nodes is below this limit, CDN is worse than CCN, and the opposite happens above this threshold.

Fig. 4(a) shows the traffic trend for a CDN with a variable number of nodes as well as cache size. When the size of the available cache is modest and equal to 5 objects per node, increasing the number of CDN nodes has limited benefits. On the other hand, when the cache storage increases, having the chance to deploy many CDN servers is clearly beneficial. Like in the previous figures, increasing the cache size to 100 objects per node does not produce relevant benefits for the objective function.

The impact of the cache size in CCN is depicted in Fig. 4(b) and 4(c) for the Abilene and the Random-Geometric Graph, respectively. As the traffic demand generated by the clients increases, the total bandwidth has a linear trend. This behavior is expected since network links are not congested, due to the presence of the caches. Also, we clearly notice that increasing the cache available in the network has limited benefits when passing from 50 to 100 objects per node. Similar observations were previously made for Fig. 3(a).

VI. Conclusion

In this paper we proposed a novel theoretical framework, based on integer linear optimization techniques, to analyze the performance of a Content-Centric Network and to provide clear comparative results with a Content-Distribution Network.

Performance bounds were derived by addressing the joint *object placement* and *routing* problem. By performing an extensive numerical analysis we discovered that: 1) The presence of a distributed cache in both the CCN and CDN architectures can have significant benefits for the QoS of the network since it makes possible to accommodate much higher traffic demands even when few objects are stored in the nodes. 2) For large caching storage, the benefits of using sophisticated *cache replacement policies* are dramatically reduced. 3) A Content-Distribution Network can provide slightly better performance than a CCN, even when the total amount of caching storage deployed in the network is exactly the same. This is possible due to the fact that a CDN provides the additional degree of freedom to choose the *location* of the distributed cache.

The previous results can be generalized to different topologies, since the trends that we obtained were similar with respect to all of them. To the best of our knowledge this is the first attempt to model and compare the performance of a CCN with respect to that of a CDN without using a simulated model.

Acknowledgment

This work was partially supported by Italian MIUR and French ANR in the framework of the PRIN Gatecom and ANR Green-Dyspan projects.

References

[1] Cisco, "Cisco Visual Networking Index," http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf, Last accessed: Mar. 2013.

[2]

[3] B. M. Leiner, V. G. Cerf, D. D. Clark, et al., "The Past and Future History of the Internet," *Communications of the ACM*, vol. 40, no. 2, pp. 102–108, 1997.

[4] G. Pallis and A. Vakali, "Insight and Perspectives for Content Delivery Networks," *Communications of the ACM*, vol. 49, no. 1, pp. 101–106, 2006.

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. of the 5th Int.l Conference on Emerging Networking Experiments and Technologies (CoNEXT '09)*. ACM, pp. 1-12, Rome, Italy, 2009.

[6] D. K. Smetters and V. Jacobson, "Securing network content," PARC, Tech. Rep., 2009.

[7] D.-h. Kim, J.-h. Kim, Y.-s. Kim, H.-s. Yoon, and I. Yeom, "Mobility Support in Content Centric Networks," in *Proc. of the 2nd ICN Workshop on Information-Centric Networking (ICN '12)*. ACM, pp. 13-18, Helsinki, Finland, 2012.

[8] I. Baev, R. Rajaraman, and C. Swamy, "Approximation Algorithms for Data Placement Problems," *SIAM Journal on Computing*, vol. 38, no. 4, pp. 1411–1429, 2008.

[9] J. Kangasharju, J. Roberts, and K. W. Ross, "Object Replication Strategies in Content Distribution Networks," *Computer Communications*, vol. 25, no. 4, pp. 376–383, 2002.

[10] S. Borst, V. Gupta, and A. Walid, "Distributed Caching Algorithms for Content Distribution Networks," in *Proc. of the 29th IEEE Conference on Information Communications (INFOCOM'10)*, pp. 1478-1486, San Diego, California, USA, 2010.

[11] T. A. Neves, L. S. Ochi, L. M. Drummond, and E. U. Célio, "Optimization in Content Distribution Networks," in *Proc. of the Int.l Conference on Engineering Optimization, (EngOpt)*, Rio de Janeiro, Brazil, June 2008.

[12] G. Pallis, K. Stamos, A. Vakali, D. Katsaros, and A. Sidiropoulos, "Replication Based on Objects Load Under a Content Distribution Network," in *Proc. of the 22nd Int.l Conference on Data Engineering Workshops (ICDEW '06)*, pp. 53, Atlanta, Georgia, USA, April 2006.

[13] T. Nguyen, C. T. Chou, and P. Boustead, "Resource Optimization for Content Distribution Networks in Shared Infrastructure Environment," in *Proc. of the Australian Telecommunications Networks Applications Conference (ATNAC 2003)*, Melbourne, Australia, December 2003.

[14] G. Rossini and D. Rossi, "Evaluating CCN Multi-Path Interest Forwarding Strategies," *Computer Communications*, vol. 36, no. 7, pp. 771–778, 2013.

[15] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice-Over Content-Centric Networks," in *Proc. of the ACM Workshop on Re-architecting the internet*, pp. 1–6, Rome, Italy, 2009.

[16] H. Yuan and P. Crowley, "Performance Measurement of Name-Centric Content Distribution Methods," in *Proc. of 7th ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, pp. 223–224, Brooklyn, NY, October 2011.

[17] M. Varvello, D. Perino, and J. Esteban, "Caesar: a Content Router for High Speed Forwarding," in *Proc. of 2nd ACM SIGCOMM Workshop on Information-Centric Networking (ICN)*, Helsinki, Finland, August 2012.

[18] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," *IEEE INFOCOM, NOMEN Workshop*, pp. 310–315, 2012.