Computer Networks 57 (2013) 3116–3127

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

From content delivery today to information centric networking

G. Carofiglio [a], G. Morabito [b,*], L. Muscariello [c], I. Solis [d], M. Varvello [e]

[a] *Bell Labs, Alcatel-Lucent, France*
[b] *CNIT RU at University of Catania, Italy*
[c] *Orange Labs, France Telecom, Italy*
[d] *Palo Alto Research Center, Italy*
[e] *Bell Labs, Alcatel-Lucent, USA*

ARTICLE INFO

Article history:
Available online 12 July 2013

Keywords:
ICN
Information centric networks
CDN
Content delivery networks

ABSTRACT

Today, content delivery is a heterogeneous ecosystem composed by various independent infrastructures. The ever increasing growth of Internet traffic has encouraged the proliferation of different architectures to serve content provider needs and user demand. Despite the differences among the technology, their low level implementation can be characterized in a few fundamental building blocks: *network storage*, *request routing*, and *data transfer*. Existing solutions are inefficient because they try to build an information centric service model over a network infrastructure which was designed to support host-to-host communications. The Information-Centric Networking (ICN) paradigm has been proposed as a possible solution to this mismatch. ICN integrates content delivery as a native network feature. The rationale is to architect a network that automatically interprets, processes, and delivers content (information) independently of its location. This paper makes the following contributions: (1) it identifies a set of building blocks for content delivery, (2) it surveys the most popular approaches to realize the above building blocks, (3) it compares content delivery solutions relying on the current Internet infrastructure with novel ICN approaches.

1. Introduction

The Internet was originally conceived to enable communication between remote hosts. Then, broadband access penetration and mobile terminal evolution have transformed the Internet into a medium to connect people with *content*: users search for information over Google, watch videos on YouTube, and share files via BitTorrent. Content dissemination has caused an exponential growth of Internet traffic volume; this trend is expected to continue in the near future and is currently driven by video-centric services.

Today "content delivery", i.e., getting content from a producer to a consumer, involves many technologies (e.g., content delivery networks (CDNs) and peer-to-peer networks), and players (e.g., CDN providers, ISPs and content providers). The evolution of content delivery has mainly been driven by market needs rather than a coherent architectural plan. Consequently, today content delivery is highly heterogeneous. In addition, solutions proposed in these domains are characterized by inefficiencies due to the fact that they try to match the content centric problem with an infrastructure which is built on top of a host-to-host communication model.

To overcome the friction caused by the difference between the content delivery problem and the host-to-host communication model, the research community has recently focused on *Information-Centric Networking* (ICN) [25,23], a novel networking paradigm that integrates content delivery as a native network feature. The rationale is to architect a network that automatically interprets, processes, and delivers content (information) independently of its location. The communication shift is realized by replacing host addresses with content names. Named data

[*] Corresponding author. Tel.: +39 095 7382355; fax: +39 095 7382397.
E-mail addresses: giacomo.morabito@dieei.unict.it, morabito.giacomo@gmail.com (G. Morabito).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.07.002

Table 1
Projects on information-centric networking.

Project	Website
AsiaFI	http://www.asiafi.net/
COMET	http://www.comet-project.org/
ANR Connect	http://www.anr-connect.org/
Convergence	http://www.ict-convergence.eu/
GreenICN	http://www.greenicn.org/
NDN	http://www.named-data.net/
NetInf	http://www.netinf.org/
PSIRP	http://www.psirp.org/
PURSUIT	http://www.fp7-pursuit.eu/
4WARD	http://www.4ward-project.eu/
SAIL	http://www.sail-project.eu/

is exchanged upon user demand, and can be cached by any network component (including routers) equipped with storage.

Since its proposal, the research community has payed an increasing interest in the ICN approach as demonstrated by the large number of research projects [2] (we report a list of relevant projects along with the links to their websites in Table 1) and and by the recent establishment of the *Information-Centric Networking Research Group* (ICNRG, see http://irtf.org/icnrg) within the Internet Research Task Force (IRTF). In this paper, we look at the basic building blocks required for content delivery and how they map to the general ICN concepts. Using this as a base we then finish off by comparing the ICN model with the current content delivery model.

The remainder of the paper is organized as follows. In Section 2 we describe existing solutions for content delivery that rely on the current Internet. More specifically, we characterize such solutions referring to the three major building blocks which can be identified in content delivery:*request routing*, *network storage*, and *data transfer*. In Section 3 we provide an overview of the approaches that ICN solutions propose for the above building blocks. A comparison of today's content delivery model versus the one proposed by ICN is the subject of Section 4. Finally, in Section 5 we will draw our concluding remarks.

2. Content delivery

In this section, we overview the low level mechanism used by state-of-the-art content delivery mechanisms. Despite the differences among the technology, their low level implementation can be characterized in few fundamental building blocks: *network storage*, *request routing* and *data transfer*, which will be addressed in Sections 2.1, 2.2, and 2.3, respectively.

2.1. Network storage

Content-delivery solutions can serve a piece of content from a set of network locations on behalf of the original content provider. We call this function *network storage*. Network storage allows improving end-user response time while reducing network traffic and server load; also, it allows increased service availability. Fig. 1 graphically shows how network storage solutions have been deployed over the years, from simple Web caching to Content Delivery Networks and ISP transparent caching. We will go over each of these solutions while analyzing their strengths and weaknesses.

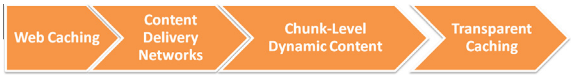

Fig. 1. Network storage evolution over the years.

Web caching [39] is the first widely adopted network storage solution. Web caching leverages a set of Web proxies augmented with caching capability. Requests arrive from the clients browsers which are configured (statically or dynamically) to use the proxy. Web caches are normally organized in a hierarchical structure so that cache misses can be resolved upstream if possible. The major limitations of Web caching are the following: (i) *scalability*: traffic overload may still arise at the top of the hierarchy in the presence of flash crowds [51]; (ii) *content consistency*: there is no coordination between Web caches and content providers; (iii) *lack of transparency*: users may need to manually configure their browsers with the proxy address.

Content Delivery Networks (CDNs) [33] were introduced to overcome the limitations of Web caching. These limitations have been highlighted by the widespread of Web services like dynamic content, video on demand and live streaming. CDNs are complex networks of servers distributed across the Internet that cooperate for content delivery. CDNs differ from Web caches as they are completely transparent to the users, and they proactively replicate content at network edge. Also, CDNs maintain a direct relationship with content providers. Furthermore, by leveraging on specific markup languages (such as the *Edge Side Include*[1] (ESI)) CDNs support the dynamic assembly and delivery of content, e.g., *dynamic content*. Recent CDNs operate on content *chunks*, i.e., fractions of a file, allowing fine-grained load balancing even for large-file content retrievals [32].

Today's CDN model is based on at least three main players: the content providers, the CDN providers and the network operators. The relations between the three are mainly based on explicit contracts with clear customer/provider roles. There are no standard mechanisms to implement content publication and delivery down to the final users. Most of the time the interconnection is based on manual configuration and based on the very specific content and CDN technology being used. As a result, the three worlds are usually distinct and evolve as silos. CDN providers are IT (Information Technology) companies and focus on optimizing their software, while network operators on managing reliable interconnection and quality of service. In some cases network operators, as Level 3, are able to offer global CDN services over their international backbone network, implementing an IT/Network convergent service. In other cases nation-wide ISPs deploy their own CDN service installing servers in regional and national PoPs to reduce traffic costs in cross exchange points. IT/

[1] http://www.esi.org.

Network convergence is however not possible for two main elements:

- Current HTTP-based IT is not able to run at an IP line card rate; as a result IT is deployed as a slow path in the content delivery chain.
- There is no protocol today to interconnect the various CDNs among them. The failure of the CDNi IETF working group http://tools.ietf.org/wg/cdni/ in providing a running protocol to realize interconnection is a consequence of the aforementioned limitations.

As a consequence the CDNs suffer from the limitations reported below. First, they have no fine-grain control on where to place their servers as this requires collaboration of multiple ISPs. Second, CDNs only serve content for a subset of applications based on their agreements with the content providers. Third, there is a lack of communication among different CDNs, i.e., CDNs managed by independent providers or autonomous CDNs within the same organization, and their interconnection would require the design of complex ad-hoc mechanisms.

The above limitations are a strong incentive for telco operators to enter the content delivery ecosystem through the deployment of ISP-operated CDNs and the integration of transparent caching into their networks. ISP-operated CDNs rely on the technology discussed above.

Transparent caching[2] is a network storage solution directly embedded into a carrier network; thus, it gives the operator full control on what, where and when to cache. Transparent caching allows using a single underlying caching infrastructure for content of different applications. Also, it does not require agreements between operators and content providers.

Despite the clear benefits, the management of transparent caching still presents various technical challenges. For example, storage management policies must handle different content types with associated requirements in terms of time-to-live, service priority etc. Furthermore, they cannot leverage content provider collaboration for content type discovery. Deep packet inspection techniques are often employed for such purpose, however, they require expensive network equipment.

Web caching, CDNs and transparent-caching account for a huge amount of storage resources across the Internet. These resources are spread across a heterogeneous set of technical solutions owned by different organizations. This heterogeneity leads to three major limitations: (i) suboptimal resource utilization, as every infrastructure is independently optimized, often regardless of the underlying transport network; (ii) increasing complexity in terms of management, especially when multiple solutions should be interconnected; (iii) data consistency, as content providers do not have direct relationships with all owners of the infrastructures where their content is stored.

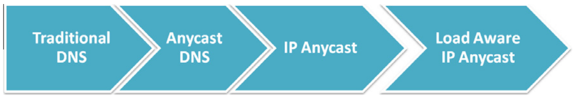

Fig. 2. Request routing evolution over the years.

2.2. Request routing

Content delivery starts by mapping a user's content request to a copy of the content. The main objective is to map the request to the best located network node with respect to the user. Name-based minimum cost routing realizes this mapping by translating a human-readable content name into the IP address of a content delivery node. We call such function *request routing*. Request routing is rather complex as the cost function to minimize depends on several factors: network topology, content availability, server conditions and network state. Fig. 2 summarizes the evolutionary steps of request routing from a DNS-based design to load-aware IP anycast.

Today, most commercial content delivery solutions exploit the Domain Name System (DNS) to implement request routing. This is a natural choice for the following reasons: (i) DNS is used by most applications, and (ii) it has an ubiquitous infrastructure densely deployed worldwide. However, the DNS was designed for name resolution, i.e., translate host names into host addresses, and not for request routing. In the content delivery market, two popular DNS-based request routing mechanisms [9] are used: in the following, we refer to them as "traditional DNS" and "anycast DNS".

Let us summarize how *traditional DNS* works. The user requests a content item through a regular URL, i.e., the human-readable content name. This request triggers a query to the user's recursive DNS which forwards the query to the authoritative DNS of the content delivery infrastructure,[3] unless the mapping <content name,IP address> is already available at the recursive DNS cache. The authoritative DNS replies with the IP address of the best content delivery node for this user request. To this aim, the authoritative DNS is constantly aware of the status of the content delivery infrastructure (e.g., load and content availability); also, it derives the network location of a user by geo-locating its recursive DNS server. The recursive DNS stores the information sent by the authoritative DNS about the best location where a certain content can be retrieved from. However, such information is not permanent: it will elapse after a time interval which is usually set to 20 seconds. It follows that recursive DNS servers can cache these replies only for a short time. Accordingly, most of the requests from the users will be handled by the authoritative DNS, which might become the bottleneck of the system.

Such usage of the DNS infrastructure is questionable [54] and has a number of limitations. First, a content delivery architecture has no control of the DNS infrastructure: recursive DNSes belong to third parties that do not provide a performance guarantee. For example, Pang et al. [31]

[2] http://blog.streamingmedia.com/the_business_of_online_vi/2010/10/an-overview-of-transparent-caching.html.

[3] This step supposes cooperation between original content provider and the content delivery infrastructure.

show that about 50% of recursive DNSes do not respect the TTL values recommended by a content delivery system. Second, DNS-based geo-localization does not work when a user is located far away from its recursive DNS server. Third, not all DNS requests represent the same user workload: a request may correspond to a short data transfer, like an HTML page download, or a long session, like video content retrieval.

Anycast DNS is another request routing mechanism that leverages the DNS. "Anycast" refers to the ability of IP routing to select the shortest path between a user and multiple endpoints associated to the same IP anycast address. Anycast DNS works as follows. A set of authoritative DNSes are identified by the same IP anycast address; such authoritative DNSes are located close to content delivery nodes and reply to DNS queries with the IP addresses of close-by nodes. This approach also suffers from the intrinsic limitations due to the misuse of the DNS, but it is better integrated with the underlying IP network.

IP Anycast can also be directly used as a request routing mechanism by addressing content delivery nodes serving the same content with the same anycast address. In this way, IP anycast would route each content request to the optimal content delivery node from an IP perspective, i.e., shortest path. This request routing design has two main advantages. First, it significantly adheres to IP routing, which provides fine grained fail over, reduced impact of DDoS attacks and improved performance. Second, it requires little or no usage of the DNS infrastructure. Unfortunately, this design lacks some expressiveness as shortest path is only one of the metrics that request routing should consider in the selection of a content delivery node. Also, any routing change that causes anycast traffic to be re-routed to an alternative anycast endpoint may cause a session reset in the middle of connection-oriented transfers (e.g., HTTP and TCP). Due to these limitations, IP anycast is not used by today's content delivery solutions.

However, the research community has already investigated how to overcome some of the above mentioned limitations. Al-Qudah et al. [6] propose *load-aware IP anycast*, a mechanism that augments IP anycast with additional information such as load at the anycast end-points, i.e., the content delivery nodes. This mechanism can be implemented using a centralized controller and a proper measurement information base at every content delivery node. Unfortunately, this approach can only target ASes with a large footprint in the country where they provide content delivery service, e.g., Tier-1 ISPs in the US. Also, the central controller represents a single point of failure as well as a possible performance bottleneck.

2.3. Data transfer

Content-delivery solutions are responsible for the transfer of a requested content item from one or multiple storage locations to the user. We call this function *data transfer*. Fig. 3 shows the evolution of data transfer over the years from client/server to HTTP-based receiver-driven. Under the *client/server* paradigm, each content request is served by a unique server that delivers the requested content using a TCP connection. TCP has been successfully designed for client/server communication, providing scalable and reliable data transfer in a "sender-based" fashion. TCP is sender-based as the sender controls bandwidth allocation while performing loss detection and congestion control.

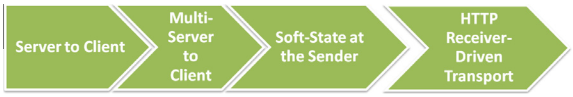

Fig. 3. Data transfer model evolution over the years.

With the introduction of content-delivery infrastructures, the data transfer paradigm evolves into a *multi-server* model. In the multi-server model, the task of the transport protocol is to coordinate the data transfer from the multiple locations to optimize resources utilization (server bandwidth and link load). When the workload is composed of short data transfers, optimizing resources utilization simply means balancing content requests across servers. Requests balancing is not sufficient when long data transfers constitute a significant portion of the overall workload. In such case, the transmission source needs to be relocated to a different server, also called sender relocation. For example, sender relocation is needed for video delivery as available network resources can drastically change during the transfer lifetime.

TCP does not naturally support sender relocation due to its connection-oriented and sender-based nature. In fact, a TCP session is identified by a 4-tuple consisting of the IP addresses and the port numbers of both the end nodes. Furthermore, for each TCP session each of the end nodes keeps updated some *state* information such as the sequence number of the next packet which is expected and the value of the congestion window. To realize sender relocation in TCP, senders need to migrate all the above information. This approach has low overhead, though it may cause a non-negligible synchronization delay; also, it becomes considerably complex when the number of sources is large. An alternative approach is to reduce, or possibly remove, the socket state at the server side (*soft-state*). Trickles [44] realizes soft-state at the sender by encoding the connection state within each packet. The sender endpoint exploits this information to drive congestion control decisions at the receiver, thus enforcing TCP friendliness.

Ideally, delegating congestion control to the receiver could solve the above mentioned issues. *Receiver-driven transport* has several advantages [28], such as mobility and multi-path support to name a few. The major drawback is that the sender has no control on bandwidth allocation and misbehaving receivers may unfairly obtain more resources. As a countermeasure, it is necessary either to use solutions like Trickles or to implement per-flow fair queuing at the sender output interface. Note that fair queuing imposes a different fairness objective with respect to sender-based TCP (max-min instead of proportional fairness). However, it has been shown that the fairness criterion has little impact at normal traffic loads [11]. The use of stateless sockets at the sender not only reduces load at the server but enables packet-level failover, fine grained load balancing and robustness to anycast re-route.

Therefore, such transport protocols appear to better fit the communication requirements of content-delivery.

Alternatively, a lightweight sender end-point can also be implemented at the application layer. This approach is motivated by the proliferation of *HTTP-based* content delivery. Among the advantages, that make HTTP the de facto transport protocol [37] for multimedia delivery, there are NAT and proxy traversal, simple user interface supported by all Web browsers and wide deployment. The success of HTTP is confirmed by the recent introduction of *Dynamic Adaptive Streaming over HTTP*,[4] already deployed in proprietary solutions. DASH specifies data segment units uniquely referenced by HTTP URL like identifiers that can be retrieved by HTTP 1.1 requests. This approach can be generalized to any kind of information available in the Internet leveraging the range requests in HTTP 1.1. Moreover, it can be used to implement robustness to sender relocation, as HTTP GET includes all the information needed to keep the server stateless or, if stateful, robust to server relocation and any-cast reroute, as proposed in [4]. However HTTP is not a transport protocol as it relies on TCP to implement error recovery, significantly increasing the transport overhead.

3. Information-centric networking

Information-Centric Networking (ICN) [2] is a novel network paradigm where communication takes place by exchanging named data instead of transmitting packets from a source to a destination. This framing makes ICN, like CDNs, put content and its retrieval at the heart of networking. Using ICN for content delivery can potentially overcome the current CDN's limitations with a new set of protocols that take advantage of the complete network infrastructure.

The details of ICN have not yet been settled. There is still plenty of debate about the general service model, architecture, and protocols that will be standardized, but there is consensus on some of the features that ICN will provide. Throughout the paper, we use the term ICN to refer to a complete system that includes the core networking technology as well as the services built immediately on top of it.

First of all, content in ICN is identified by a *name*; accordingly, content requests are simply addressed to the content name. Network entities use this name to route content requests towards one of the machines that hosts an authorized copy of the desired content. It follows that ICN will support some form of *name-based routing* and *forwarding*, where a packet is addressed to content name rather than a geographic location, e.g., the IP address. Similarly to modern CDNs, the selection of the specific host (or subset of hosts) where the request message is forwarded may depend on several factors, e.g., the distance from the requesting user, the current server load, and the traffic load in the network.

Content authentication and protection is another key feature provided by most ICN solutions. In the Internet, most of the effort in the security domain has been put on authenticating sender and receiver and protecting the connection between them. Such approach has demonstrated strong limits:

- It does not prevent denial of service attacks which can cause the unavailability of a certain piece of content.
- It does not give guarantees about the authenticity of the piece of content which is being retrieved (just think of the P2P case in which it is impossible to realize whether a given content is authentic before the download is complete).

Conversely, in ICNs the content is authenticated as to be protected against alteration or eavesdropping. Accordingly, an ICN network only permits the circulation of genuine copies of the content. Furthermore network storage makes denial of service attacks less effective. In fact, given that several replicas of requested contents are available in the network, load of content servers is dramatically reduced.

ICN provides *in-network caching*; ICN routers are expected to be equipped with a cache where they can store (part of) the content they relay. Upon reception of a a request message, routers check in their cache whether a copy of the desired content is available. If this is the case, they stop propagating the request message and send the desired content to the requesting user. Otherwise, they just forward the request message based on a lookup in the local forwarding table.

Finally, in most designs ICN will support the *publish/subscribe* service model: users generates *requests* for a certain piece of content and the network provides the user with such content when this becomes available. Note that according to this approach users may request a piece of content even when the content is not yet available.

In the remainder of this section, we characterize ICN by the same three fundamental functions described for content delivery today: network storage, request routing and data transfer.

3.1. Network storage

The ICN paradigm exploits the opportunity of storing (popular) content inside the network, i.e., *in-network caching*. Indeed, all ICN solutions envisions routers equipped with a local cache in which they can store part of the data flowing through them. Before forwarding an interest message[5] requesting a certain piece of data X (let us denote such a message as $Q(X)$) to the next hop, they check whether X is currently stored in the local cache. If this is the case, then they send the requested piece of data X to the interested user and drop the interest message. Otherwise they pass the data request to the next hop towards the server containing the requested data.

For example, consider the exemplary network shown in Fig. 4. Node B issues an interest message for given piece of content X. Such content has been published by node F and

[4] 3GP/MPEG DASH, http://www.3gpp.org/ftp/Specs/html-info/26247.htm.

[5] In the following we use the terms *interest message* and *request message* interchangeably.

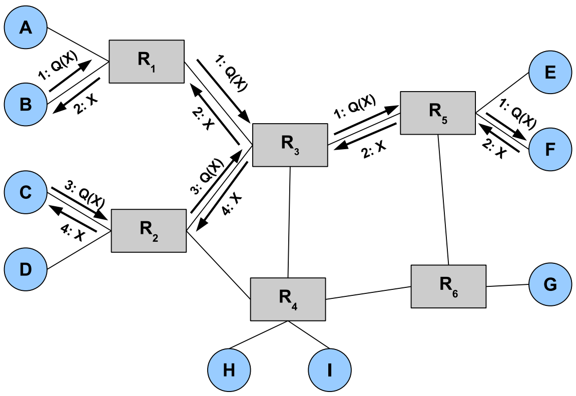

Fig. 4. An example of in-network caching.

therefore, the request message $Q(X)$ traverses routers R_1, R_3, and R_5. Node F receives the interest message issued by node B and sends it the desired piece of content X. Content data X will traverse nodes R_5, R_3 and R_1 back which will store local copies of such piece of content in their caches.

Then, suppose that node C generates an interest message for the same piece of content X. The related request message $Q(X)$ will traverse routers R_2 and R_3. The latter – router R_3 – realizes that there is a copy of X in its cache and therefore, it does not forward the interest message to the next hop towards F. Instead R_3 sends the content X to node C.

It is obvious that exploiting in-network caching, network performance may be improved significantly in terms of network resource utilization, data availability, and data delivery delay [7]. In fact, as a consequence of the spatial and temporal locality which characterizes content requests in the Internet [5,10], it is likely that the same content is requested several times by users geographically close to each others within a short time interval.

In the context of in-network caching, the major problems to address are related to the selection of the locations where to put caches and of the specific content items to store in the different caches. In general, two opposite approaches can be distinguished to address such problems: *coordinated* and *uncoordinated* [40].

In the *coordinated* case, routers exchange information to achieve a better estimation of the popularity of contents and to avoid storage of too many copies of the same content. For what concerns the latter points, consider the exemplary network depicted in Fig. 4 and suppose, as an extreme case, that a very popular piece of content is stored in all routers. It is evident that the copy stored in router R_3 is useless given that interests from the users will be satisfied by their access router.

In the *uncoordinated* case, each cache operates autonomously and thus the so called *Transparent En-Route Caching* policy is used in which each router in the end-to-end path between the interested user and the content source decides whether to cache transiting pieces of content and which cache replacing strategy without interacting with other routers. For what concerns the replacing strategy, although in most scenarios in which the popularity of objects does not change very much over a specific time period the *least frequently used*[6] approach would achieve the highest performance [36]; most proposed solutions adopt the *least recently used.*[7]

Obviously, the effectiveness of in-network caching is higher when coordinated-type solutions are utilized. However the overhead required to manage the coordination between caches may become extremely high. As a result, at present time uncoordinated solutions are adopted more frequently.

Note however, that between the two above opposites – the *coordinated* and *uncoordinated* approaches – described above several trade-offs are possible. Relevant examples include [49,14].

In the context of in-network caching an interesting mechanism is proposed in [35] to reduce redundancy in caches. The basic idea is to identify the content stored in the cache through the content itself instead that through its name. In this way the same sequence of bytes can be used by different sessions increasing efficiency of the utilization of network resource.

A large research effort is being devoted to the theoretical analysis of in-network caching as well. In fact, attempts to model the behavior of in-network caching trees and strategies can be found in [38,29,18,41] whereas in [50] a Markovian model capturing the essential dynamics of in-network caching is defined as a way to calculate the survival time of information items in an ICN.

In the hype of enthusiasm attracted by in-network caching a few dissonant voices have been raised. In fact, in [34] authors question how far in-network caching effectiveness can go taking into account the current limitations of memory used for content storage. Instead, in [20] authors start their discussion from theoretical results demonstrating that the effectiveness of caching increases logarithmically with the size of the caches. Based on the above result, they conclude that it is not convenient to radically change the Internet architecture just to support in-network caching. In fact, the logarithmic increase in the effectiveness of in-network caching with the size of the caches cannot cope with the exponential growth of Internet content.

3.2. Request routing

ICN enables "location-independent" access to content, which means that users can retrieve content without knowing its location. It follows that content identifiers are location-independent; this calls for appropriate mechanisms to route content requests issued by users towards the most appropriate *serving nodes*, i.e., nodes that store a copy of the desired content.

A possible approach to satisfy the above need is to move the DNS-like functionality inside the network. The resulting solutions, which can be considered as evolutions of the CDN approaches described in Section 2.2, rely on an

[6] The cache stores the pieces of content which have been requested most frequently in a given time interval.

[7] The cache stores the pieces of contents that have been requested most recently.

appropriate service offered by the network that maps the identifier of the desired content into the address (location-dependent) of the most appropriate serving node. Once such mapping operation is completed, the request can be routed and forwarded by using any traditional routing and forwarding mechanism.

Accordingly, this approach has three conceptual routing phases as explained in [2]

- Routing the request message to an appropriate serving node where the content name is translated into one or multiple addresses of nodes where a copies of such content are available.
- Routing the request message to such node(s).
- Routing the data from such node(s) to the requester.

For example, in [43,53] the mapping between content identifier and a location-dependent identifier is executed by a set of nodes that constitute a *rendezvous network*. When a node wants to publish some content and thus serve as persistent storage for it, it will inject the pair <content, location> into the rendezvous network. Such information will be distributed within the rendezvous network using mechanisms like those proposed in DONA [25]. A node interested in a certain content will query the rendezvous network that will return the sequence of links (each with its own identifier) which must be traversed by the content request to reach one of the serving nodes, thus realizing "source routing" [24]. The selection of the routing hops is done according to a certain metric which can be decided at each time.

The main advantage of this approach is that it can be easily integrated into the existing network infrastructure. However, this comes at the price of a very high *stretch factor*: this is the ratio between the length of the path a request message goes through and the shortest path between source and destination. In DONA the stretch factor can be high because a content request must reach a node in the rendezvous network and then must be returned back to the requesting node before it can be sent towards the appropriate serving node. Such high stretch factor results in inefficient use of network resources and long delay. In addition, this approach suffers from additional overhead due to source routing: each packet header has to contain the identifiers for all links of the path it traverses [24].

In order to solve the above mentioned issues, researchers have proposed extending routing and forwarding tables in order to support location independent content identifiers as input instead of location-dependent network addresses. In this way, a content request only needs to contain the content name avoiding the overhead caused by source routing. In addition stretch is eliminated, i.e., stretch factor equals 1. Unfortunately, location-independent identifiers are hard to aggregate both in forwarding and routing tables causing an explosion in their sizes.

In order to achieve aggregation, Jacobson et. al [23] propose to use hierarchical human-readable names to address content items; content name has several *components* delimited by a character, e.g., `/Research/PAPERS/PaperA.pdf`. A content request propagates toward potential data using Longest Prefix Matching (LPM), as in IP: routers use LPM to select the entry from the local forwarding table that shares the longest prefix with the content name contained in a packet header. For example, a content request for `/Research/PAPERS/PaperA.pdf` might match a *content prefix* such as `/Research/PAPERS/*`.

Building a router that supports high speed forwarding on hierarchical names, a *content router*, has two main challenges. First, a content router has to perform LPM for at least the same amount of packets processed today by a high-end router, assuming forwarding tables that are several orders of magnitude larger [34,8]. Second, its forwarding table contains content prefixes that have a large number of components and unlimited characters per component.

In order to solve the above challenges, Varvello et. al [52] propose two main ideas. First, distribute the forwarding table across line cards in a router in order to maximize the overall size. The downside of this approach is a possible increase in switching operations that is absorbed by additional switch fabrics as commonly done in commercial routers [15]. Second, LPM is designed to be as independent as possible from the length of content names: this is achieved by adapting the distributed Bloom filters approach proposed in [48] to content names. Numerical results show that speed up to 40 Gbps assuming forwarding tables with up to one billion content prefixes are possible [52]. An experimental evaluation to confirm such numerical results is undergoing.

Flat naming is an alternative to hierarchical naming that uses a flat name-space and a resolution step, either explicit as in [12] or integrated with forwarding as in [45]. A major critique to flat naming is related to its *scalability*. In fact, simple flat naming does not support aggregation intrinsically, which may cause an explosion in the size of the forwarding/routing tables. In [21], the authors propose *explicit aggregation* as a mechanism to reduce the size of forwarding/routing tables.

The basic idea of explicit aggregation is to concatenate flat names as follows. Suppose A, B, and C are unique flat names that identify the publisher of a certain multimedia content, the multimedia content itself and a small portion of such content, respectively. Then it is possible to build an explicit aggregation as follows $A.B.C$. The assumption is that by forwarding a request for $A.B.C.$ towards A, a forwarding table's entry towards B and then C will be eventually found. It follows that forwarding of content requests will be based on a *deepest first* mechanism. If a forwarding table has an entry for A and another for B, then forwarding will consider the entry towards B; if there is a forwarding entry for C as well, then this is the one that will be considered. Note that explicit aggregation is more flexible than the aggregation which is intrinsic in hierarchical naming. For example, consider the case in which the same multimedia content B has been published by two entities A_1 and A_2. Then it is possible to refer to the same content B as $A_1.B.C$ as well as $A_2.B.C$. This would be impossible with hierarchical naming. In fact, if hierarchical naming is used $A_1.B.C$ and $A_2.B.C$ would refer to two different pieces of content given that the names at the highest level of the hierarchy are different. For example, `Alice/myFirstDocument.pdf` and `Bob/myFirstDocument.pdf` are obviously two different pieces of contents.

To conclude, let us observe that the naming scheme (either hierarchical or flat) should include information about the *principal*, i.e., the publisher/author of the content [25], in order to verify the origin of a piece of content. It follows that both naming schemes can support routing aggregation based on the principal information.

3.3. Data transfer

A major responsibility of the transport layer is to support end-to-end reliability and to implement congestion control. Support of reliability is not particularly difficult in ICNs. In fact, it is the receiver which sends requests of specific pieces of data to potential information sources and therefore, simple mechanisms can be implemented in which interested receiver retransmits requests of pieces of content which have not been received (correctly).

Congestion control, instead, represents a complex problem and proposed solutions cannot be based on the same philosophy adopted by TCP for the following reasons:

- In a TCP session all data is transmitted by one sender to one receiver (and viceversa). In ICNs different pieces of the same content can be retrieved by different servers. It follows that congestion control must be *receiver-driven*, like we have already observed in Section 2.
- In traditional TCP/IP networks the resource to be shared by different sessions was mostly *bandwidth*. In ICNs also *cache space* must be considered and fairness metrics should account for it as well.

It follows that new appropriate solutions are needed. For congestion control the major design objectives to pursue are the following [13]:

- **Efficiency**: Congestion control solutions cannot rely on explicit congestion notification from the underlying layers and therefore, must recognize congestion occurrences by observing the network from the outside. To avoid network collapse in such contexts, congestion control schemes must be conservative and reduce the transmission rate whenever a sign of a congestion occurs. This may result in low utilization of available network resources.
- **Fairness:** Available network resources must be shared fairly between competing sessions. This is complex because intermediate network nodes cannot maintain state information about all active sessions traversing them.

In its first implementations the *Content-Centric Networking* (CCN) platforms [23] did not provide the transport layer functionality: such operations are transferred to the application layer. To fill this gap the *Interest Control Protocol* (ICP) has been proposed in [13]. ICP is a *receiver-driven* protocol in which reliability is achieved by letting the receiver re-transmit a request message if the corresponding piece of data is not received within a certain time interval. As far as the congestion control is concerned, ICP implements an additive increase, multiplicative decrease (AIMD) discipline. In fact, the receiver utilizes a window specifying how many request messages the receiver can introduce in the network (without receiving the corresponding data). Upon receiving a data packet the window is increased additively. If a certain piece of data is not received within a certain time interval then the window is reduced multiplicatively. Note that by appropriately setting the values of key parameters ICP behavior resembles that of the congestion avoidance as well as fast retransmit and fast recovery algorithms implemented by TCP. An analytical model of ICP has been derived for evaluating its performance in terms of fairness and throughput.

In [42] a new transport layer protocol named Information-Centric Transport Protocol (ICTP) is proposed for the CONET networks envisioned within the CONVERGENCE project [16]. ICTP is receiver-driven such as ICP however, it uses holes in the reception of packets to detect incoming congestion in the network and replicates at the receiver side the same algorithms utilized by TCP Reno (slow start, congestion avoidance, fast retransmit and fast recovery). ICTP has been implemented and tested in real CCNx networks.

The above solutions can be considered as an adaptation of the well known TCP protocol to a receiver-driven approach. Such an approach to the congestion control problem is not suitable for solutions based on rendezvous nodes (such as PSIRP). In fact, in this case there is no direct connection between the sender and receiver. Accordingly, novel solutions are requested to detect congestion and control the rate of traffic flowing in congested areas. The two above issues are addressed by the Traffic and Congestion Control (TCC) protocol proposed in [26]. In TCC congestion is detected through packet losses and increased round trip times, whereas rate regulation is achieved through stochastic fairness queuing (SFQ) and control of the rate with which interest messages are sent.

From the discussion above it follows that several research issues are still open [27]. Since requests can be aggregated by intermediate routers,

- the transfer of a given piece of content can occur to satisfy multiple requests;
- the source of a given piece of content cannot estimate the number of requesters.

This calls for a new way to measure fairness which goes beyond the concept of flows [30].

Also, alternative approaches should be investigated which move congestion control from the end-nodes to the intermediate routers. A router-driven congestion control could achieve high effectiveness given that routers are in the best position for early detection of upcoming congestions. However, they would violate the end-to-end semantic of the transport protocol which leads to several problems [3].

Finally. in the last two decades there has been a large research effort devoted to fix the problem arising by the adoption of TCP congestion control in application domains characterized by high delay-bandwidth product, high loss probability, nodes mobility and link asymmetry (see [55,3,22], for example). Recently, proposals for the extension of ICN to such application domains have appeared

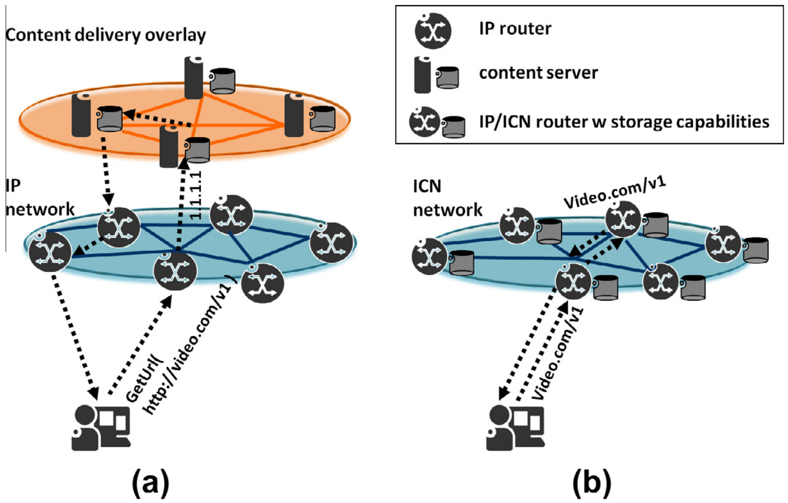

Fig. 5. Content delivery over (a) current Internet infrastructure and (b) an information-centric network.

(see [19,17,46], for example). Accordingly, it is extremely important to learn the lessons coming from research related to TCP.

4. ICN versus the current content delivery model

Content distribution is reshaping the global Internet landscape: multiple content delivery infrastructures are built on top of the traditional network architecture, forming a complex content delivery model. In this model, three main actors play a role: content delivery network (CDN) operators, content providers and ISPs. CDN providers (such as Akamai, Limelight and Level 3) develop and operate proprietary technologies, offered as a service to both content providers and small ISPs. Conversely, large ISPs deploy their own content delivery architectures to better exploit their infrastructure and pursue new revenue opportunities. To this aim, large ISPs acquire a plethora of content delivery solutions (traffic localization, application acceleration and transparent caching) from multiple vendors.

In this section, we compare today's content delivery model with the one proposed by ICN depicted in Fig. 5(a) and (b), respectively. More specifically, in Fig. 5(a) we represent the current content delivery model which relies on an overlay network of servers providing content-aware functionality over a content-oblivious infrastructure. Whereas in Fig. 5(b) we represent the information-centric networking model in which content-aware functionality is provided within the communication infrastructure. In comparing the two models, we use the following metrics: *heterogeneity*, *efficiency*, *reactivity*, and *granularity*.

4.1. Heterogeneity

The evolution of content delivery has mainly been driven by market needs rather than a coherent architectural plan. Consequently, today the content delivery ecosystem is highly heterogeneous. This heterogeneity limits and sometimes prevents the interoperability of different infrastructures. Interoperability is both a technical and operational issue [33]. On the one hand, the diversity of solutions available requires the design of ad-hoc mechanisms to allow communication among different infrastructures. On the other hand, large scale operational management becomes challenging when it involves multiple entities, often provisioned with different resources and optimized for different performance goals. The presence of many market players (CDN providers, ISPs, content providers) makes this management task further complex. The interconnection among different content delivery infrastructures is currently under investigation in the CDNi working group at IETF.[8]

Most ICN designs advocate the usage of widely distributed storage, i.e., storage capabilities embedded in the network, as shown in Fig. 5(b). A unique pervasive storage infrastructure allows overcoming the heterogeneity of existing network storage solutions and the related management complexity. ICN does not claim that storage should be placed everywhere (e.g., storage is not effective above a bandwidth bottleneck [29]); however, having the flexibility to optimally allocate storage resources where needed is a clear advantage. Also, a uniform storage infrastructure may enhance data consistency across the network, though posing the management challenges of highly distributed systems.

4.2. Efficiency

Content delivery infrastructures are mostly deployed as service overlays over IP. This separation between the transport network and the content delivery infrastructure often leads to suboptimal utilization of network resources. Bandwidth and storage are independently managed by each service-dedicated overlay with no or little knowledge of the

[8] CDNi WG, http://www.ietf.org/dyn/wg/charter/cdni-charter.

variations in the underlying network conditions. As an example, consider load balancing of requests between content delivery servers in the configuration reported in Fig. 5(a). Today, request routing performs server selection based only on the information available in the content delivery overlay (e.g., server load, content availability). Optimal server selection can only be accomplished by accounting also for network related information, e.g., path length and network congestion. Moreover, traffic engineering at the transport network might be in conflict with request routing decisions, causing potential instabilities. The lack of cooperation among content delivery infrastructures can lead to inefficient storage utilization too: the same content may be replicated multiple times in nearby servers that belong to different infrastructures.

ICN proposes a novel receiver-driven transport layer, where the receiver drives the data transfer through consecutive queries of the chunks that compose a content item. Each chunk request is independently routed to the best available serving node according to a given metric (e.g., least loaded or closest). Thus, coordination among data sources and sender relocation mechanisms are not required. Also, ICN's transport layer is tightly coupled with storage management and request routing. It follows that bandwidth and storage allocation can be jointly optimized [29], leading to a more efficient utilization of network resources than in today's content delivery solutions, where those resources are independently managed. In this context, it becomes fundamental to understand the complex interplay between data transfer and network storage and to design resource allocation mechanisms accordingly.

4.3. Reactivity

In presence of sudden changes in traffic demand or popularity, a dynamic reconfiguration of the content delivery infrastructure is required. Content replicas might have to be reorganized and request routing information should be properly updated. Content delivery architectures can be designed with a desired internal level of reactivity. However, they have little or no control on the responsiveness of the request routing mechanism, as this is mostly delegated to third parties DNS servers.

ICN integrates request routing as a native network feature via name-based routing. Name-based routing adheres to the network topology, thus providing high reactivity. Therefore, it has potential to overcome most limitations of DNS-based request routing mechanisms. Clearly, scalable name-based routing inherits the intrinsic complexity of minimum cost routing problems, and it still remains an open issue.

4.4. Granularity

Content delivery infrastructures associate content requests to their nodes with a coarse granularity. Specifically, group of users behind the same recursive DNS are seen as a single user; also, content requests are treated independently of the content size. This coarse granularity depends both on the intrinsic limitations of DNS-based request routing and on the connection-oriented nature of TCP/IP transport. In order to achieve efficient utilization of the available network resources and high user Quality of Experience, request routing decisions for each user's request should be constantly evaluated. Ideally, each user request for a set of bytes (chunk) should be routed to the best content delivery server at the time of the request. However, engineering a per-request routing mechanism under the current host-name resolution system is not scalable and would undermine the global robustness of the DNS. In addition, the relocation of the data transfer source in a connection-oriented communication requires the use of adhoc techniques, unless the problem is handled by the application itself.

A content item is identified by a globally unique reference: the content name (e.g., URI-like identifiers). The content name allows direct access to content, regardless of its temporary location. The ability to reference a chunk of the whole content item through its name enables fine-grained resource control at the price of a uniform common naming convention.

5. Conclusion

The increase of digital content generation and dissemination over the Internet has led to the creation of a complex content delivery ecosystem. Despite the various implementations, we argue that content delivery is realized via three major communication primitives: *request routing*, *network storage* and *data transfer*. In the paper, we critically analyze how each communication primitive is realized in the content delivery ecosystem, and illustrate several limitations of current approaches. We observe that the long-term sustainability of the content delivery ecosystem is undermined by technology heterogeneity, inefficient resource utilization, poor reactivity and coarse granularity in management operations.

Information-Centric Networking (ICN), a novel networking paradigm that integrates content delivery as a native network feature, promises to overcome most of the described limitations. To corroborate this hypothesis, we derive the essential building blocks of ICN and analyze their role in the content delivery ecosystem. Our analysis suggests that ICN has the potential to address content delivery requirements in the long term. However, ICN also raises important research challenges to investigate.

The design of name-based routing protocols is a first significant technical challenge for ICN. In fact, replacing host addresses with content names means expanding the size of the addressable network elements by several orders of magnitude. Today's hardware and software technology are not yet ready for such an evolution. Research is needed both on the scalability of current routing operations and on naming schemes that favor novel forms of aggregation. Also, similarly to the IP routing model, we do expect ICN to establish connectivity according to novel business models far different from today's carrier-based relations. Such business relations will strongly impact the overall routing infrastructure that we expect to change accordingly [1].

The design of simple and effective resource management mechanisms is another fundamental technical challenge for ICN. Resource management translates into the need for efficient transport protocols and storage management mechanisms, namely flow and congestion control, load balancing, cache replacement and coordination policies. We believe that an important research agenda has to be addressed before ICN can shoulder the responsibility to enable a long-term evolution of content delivery networks. At the same time, rethinking the network architecture and its communication model around content appears a necessary step to guarantee the viability of the whole content delivery chain.

Finally, let us note that in this paper we have left a few fundamental issues out of the discussion. More specifically, we have not focused on the high level architectures proposed so far for ICN because a good overview of existing literature on this topic is already available, i.e., [2]. Furthermore, the security aspects have been neglected because the related discussions would need a dedicated study on the subject. Interested readers can refer to the analyzes reported in [47,21].

Acknowledgments

We would like to thank the anonymous referees for the helpful and constructive comments. The work of GM for this paper was performed in the context of the FP7/NICT EU-JAPAN GreenICN project.

References

[1] P.K. Agyapong, M. Sirbu, Economic incentives in information-centric networking: implications for protocol design and public policy, IEEE Communications Magazine 50 (12) (2012) 18–26.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, IEEE Communications Magazine 50 (7) (2012) 26–36.

[3] I.F. Akyildiz, G. Morabito, S. Palazzo, Research issues for transport protocols in satellite ip networks, IEEE Personal Communications 8 (3) (2001) 44–48.

[4] Z. Al-Qudah, S. Lee, M. Rabinovich, O. Spatscheck, J. Van der Merwe, Anycast-aware transport for content delivery networks, in: Proc. of ACM WWW 2009, 2009.

[5] V. Almeida, A. Bestavros, M. Crovella, A. Oliviera, Characterizing reference locality in the www, in: Proc. of IEEE PDIS 1996, December 1996.

[6] H.A. Alzoubi, S. Lee, M. Rabinovich, O. Spatscheck, J. Van der Merwe, Anycast CDNS revisited, in: Proc. of ACM WWW 2008, 2008.

[7] A. Anand, A. Gupta, A. Akella, S. Seshan, S. Shenker, Packet caches on routers: the implications of universal redundant traffic elimination, in Proc. of ACM SigComm August 2008, 2008.

[8] Ashok Narayanan and David Oran, NDN and IP Routing Can It Scale? http://trac.tools.ietf.org/.

[9] A. Barbir, B. Cain, R. Nair, O. Spatscheck, Known Content Network (CN) Request-Routing Mechanisms, RFC 3568 (Informational), July 2003.

[10] P. Barford, A. Bestavros, A. Bradley, M. Crovella, Changes in web client access patterns: characteristics and caching implications, Technical report, Boston University, TR-1998-023, 1998.

[11] T. Bonald, J. Roberts, Scheduling network traffic, ACM SIGMETRICS Performance Evaluation Review 34 (2007) 29–35.

[12] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, S. Shenker, ROFL: routing on flat labels, in: SIGCOMM'06 Pisa, Italy, September 2006.

[13] G. Carofiglio, M. Gallo, L. Muscariello, ICP: design and evaluation of an interest control protocol for content-centric networking, in: Proc. of IEEE INFOCOM NOMEN 2012, April 2012.

[14] W.K. Chai, D. He, I. Psaras, G. Pavlou, Cache less for more in information centric networks, in: Proc. of IFIP Networking 2012, May 2012.

[15] Cisco CRS-1. http://www.cisco.com.

[16] A. Detti, N. Blefari-Melazzi, S. Salsano, M. Pomposini, Conet: a content-centric inter-networking architecture, in: Proc. of ACM SIGCOMM – ICN 2011, August 2011.

[17] A. Detti, A. Caponi, N. Blefari-Melazzi, Exploitation of information-centric networking principles in satellite networks, in: Proc. of IEEE ESTEL 2012, October 2012.

[18] C. Fricker, P. Robert, J. Roberts, N. Sbihi, Impact of traffic mix on caching performance in a content-centric network, in: Proc. of IEEE INFOCOM NOMEN 2012, April 2012.

[19] L. Galluccio, G. Morabito, S. Palazzo, Caching in information centric satellite networks, in: Proc. of IEEE ICC 2012, June 2012.

[20] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, J. Wilcox, Information-centric networking: seeing the forest for the trees, in: Proc. of ACM HotNets'11, November 2011.

[21] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, S. Shenker, Naming in content-oriented architectures, in: Proc. of ACM SIGCOMM – ICN 2011, August 2011.

[22] K. Harras, K. Almeroth, Transport layer issues in delay tolerant networks, in: Proc. of IFIP Networking 2006, May 2006.

[23] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, R. Braynard, Networking named content, in: Proc. of ACM CoNEXT 2009, December 2009.

[24] P. Jokela, A. Zahemszky, C.E. Rothenberg, S. Arianfar, P. Nikander, LIPSIN: line speed publish/subscribe inter-networking. in: Proc. of ACM SIGCOMM 2009, August 2009.

[25] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K.H. Kim, S. Shenker, I. Stoica, A data-oriented (and beyond) network architecture, in: Proc. of ACM SigComm 2007, August 2007, pp. 181–192.

[26] V. Koptchev, V. Dimitrov, Traffic and congestion control in a publish/subscribe network, in: Proc. of CompSysTech'10, June 2010.

[27] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, ICN research challenges, in: IETF 86 Proceedings, March 2013.

[28] A. Kuzmanovic, E.W. Knightly, Receiver-centric congestion control with a misbehaving receiver: vulnerabilities and end-point solutions, ELSEVIER Journal of Computer Networks 51 (2007) 2717–2737.

[29] L. Muscariello, G. Carofiglio, M. Gallo, Bandwidth and storage sharing performance in information centric networking, in: Proc. of ACM SIGCOMM – ICN 2011, August 2011.

[30] B. Ohlman, Report from ICNRG interim meeting. In IETF 86 Proceedings, March 2013.

[31] J. Pang, A. Akella, A. Shaikh, E. Krishnamurthy, S. Seshan, On the responsiveness of dns-based network control, in: Proc. of ACM IMC 2004, 2004.

[32] K. Park, V.S. Pai, Scale and performance in the coblitz large-file distribution service, in: Proc. of USENIX NSDI 2006, 2006.

[33] A.-M.K. Pathan, Utility-oriented internetworking of content delivery networks, PhD thesis, Department of Computer Science and Software Engineering, The University of Melbourne, Australia, 2009.

[34] D. Perino and M. Varvello, A reality check for content-centric networking, in: Proc. of ACM SIGCOMM – ICN 2011, August 2011.

[35] D. Perino, M. Varvello, K.P. Puttaswamy, ICN-RE: redundancy elimination for information-centric networking, in: Proc. of ACM SIGCOMM – ICN 2012, August 2012.

[36] S. Podlipnig, L. Bszrmenyi, A survey of web cache replacement strategies, ACM Computing Surveys 35 (4) (2004) 374–398.

[37] L. Popa, A. Ghodsi, I. Stoica, HTTP as the narrow waist of the future internet, in: Proc. of ACM SIGCOMM Hotnets 2010, 2010.

[38] I. Psaras, R.G. Clegg, R. Landa, W.K. Chai, G. Pavlou, Modelling and evaluation of ccn-caching trees, in: Proc. of IFIP Networking 2011, May 2011.

[39] M. Rabinovich, O. Spatschek, Web Caching and Replication, Addison-Wesley Longman Publishing Co., Inc.,, 2002.

[40] E.J. Rosenweig, J. Kurose, Breadcrumbs: efficient, best-effort content location in cache networks, in: Proc. of IEEE INFOCOM 2009, April 2009.

[41] D. Rossi, G. Rossini, On sizing CCN content stores by exploiting topological information, in: Proc. of IEEE INFOCOM NOMEN 2012, April 2012.

[42] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, N. Blefari-Melazzi, Transport-layer issues for information centric networks, in: Proc. of ACM SIGCOMM ICN 2012, August 2012.

[43] M. Sarela, T. Rintaaho, S. Tarkoma, RTFM: publish/subscribe internetworking architecture, in: Proc. of ICT-MobileSummit 2008, June 2008.

[44] A. Shieh, A.C. Myers, E.G. Sirer, Trickles: a stateless network stack for improved scalability, resilience, and flexibility, in: Proc. of USENIX NSDI 2005, 2005.

[45] A. Singla, P.B. Godfrey, K. Fall, G. Iannaccone, S. Ratnasamy, Scalable routing on flat names, in: Co-NEXT'10 PA, USA, 2010.

[46] V.A. Siris, C.N. Ververidis, G.C. Polyzos, K.P. Liolis, Information-centric networking (icn) architectures for integration of satellites into the future internet, in: Proc. of IEEE ESTEL 2012, October 2012.

[47] D. Smetters, V. Jacobson, Securing network content, Technical report TR-2009-1, 2009.

[48] H. Song, F. Hao, M.S. Kodialam, T.V. Lakshman, IPv6 lookups using distributed and load balanced bloom filters for 100gbps core router line cards, in: INFOCOM'09, Rio de Janeiro, Brazil, 2009.

[49] V. Sourlas, P. Flegkas, L. Gkatzikis, L. Tassiulas, Autonomic cache management in information-centric networks, in: Proc. of IEEE/IFIP NOMS 2012, April 2012.

[50] V. Sourlas, G.S. Paschos, P. Mannersalo, P. Flegkas, L. Tassiulas, Modeling the dynamics of caching in content-based publish/subscribe systems, in: Proc. of ACM SAC 2011, March 2011.

[51] A. Vakali, G. Pallis, Content delivery networks: status and trends, IEEE Internet Computing 7 (2003) 68–74.

[52] M. Varvello, D. Perino, J. Esteban, Caesar: a content router for high speed forwarding, in: ICN'12, Helsinki, Finland, August 2012.

[53] K. Visala, D. Lagutin, S. Tarkoma, LANES: an inter-domain data-oriented routing architecture, in: Proc. of ACM ReArch'09, December 2009.

[54] P. Vixie, What DNS is not, ACM Queue 7 (10) (2009). 10–10.

[55] G. Xylomenos, G.C. Polyzos, P. Mahonen, M. Saaranen, TCP performance issues over wireless links, IEEE Communications Magazine 39 (4) (2001) 52–58.

Giovanna Carofiglio received the Dr Ing degree in telecommunication engineering and in electronic and telecommunication engineering, both from Politecnico di Torino, Italy, in 2004, and the PhD in telecommunication engineering jointly from Politecnico di Torino and Telecom Paris- Tech, Paris, France, in 2008. Her graduate reseach focused on stochastic analysis of wired and wireless networks and has been performed at Politecnico di Torino and at Ecole Normale Superieure (ENS Ulm) in the INRIA-TREC group. Since July 2008, she is with Alcatel-Lucent Bell Labs, Paris working on design and performance evaluation of communication networks. She is a member of the IEEE.

Giacomo Morabito received the *laurea* degree in Electrical Engineering and the PhD in Electrical, Computer and Telecommunications Engineering from the Istituto di Informatica e Telecomunicazioni, University of Catania, Catania (Italy), in 1996 and 2000, respectively. From November 1999 to April 2001, he was with the Broadband and Wireless Networking Laboratory of the Georgia Institute of Technology as a Research Engineer. Since April 2001 he is with the Dipartimento di Ingegneria Elettrica Elettronica e Informatica of the University of Catania where he is currently Associate Professor. He serves (or served) in the editorial boards of Computer Networks (ELSEVIER), IEEE Wireless Communication Magazine (IEEE), and Wireless Networks (WILEY). His research interests focus on analysis and solutions for the networks of the future.

Luca Muscariello is a senior expert at Orange Labs, France Telecom, Paris, France. He graduated in Telecommunications Engineering from Politecnico di Torino, Italy, in 2002. He holds a PhD in Telecommunications Engineering and his graduate research was performed in the area of Internet traffic measurement, characterization and modeling at Politecnico di Torino, France Telecom R&D in Paris and at VTT in Helsinki. During 2006, he was a post-doc fellow in France Telecom R&D, working on performance evaluation of wireless networks. His current research interests include simulation and analytical modeling of wired and wireless networks. He is a member of the IEEE and of the ACM.

Ignacio Solis develops network solutions for challenged networks. At PARC (Palo Alto Research Center) Ignacio is part of the CCN (Content-Centric Networks) team and has led the work on content-based approaches for disruption-tolerant network (DTN) in various DARPA programs. In addition to designing APIs, routing and forwarding for content-based networks Ignacio has developed multiple tactical applications that use the content paradigm, including mobile and server-less applications. Ignacio's experience includes work on protocol design for heterogeneous networks and prototyping architectures. He has also worked on minimizing delay for data aggregation and optimized mapping in sensor networks. He has expertise in ad hoc networking, multicast routing and collision-free wireless MAC protocols. Ignacio received his PhD in Computer Engineering from the University of California, Santa Cruz. He did his MSc from the University of Southern California and his BS from the University of Costa Rica.

Matteo Varvello received the Research Master's degree in network and distributed systems from EURECOM, Sophia Antipolis, France, in 2006; the MSc (cum laude) degree in networking engineering from Polytechnic of Turin, Turin, Italy, in November 2006; and the PhD degree in computer science from Telecom Paris, Paris France, in December 2009. He has been a Member of Technical Staff at Bell-Labs (Holmdel, New Jersey) since January 2010. His current research interests include software-defined networking, content-centric networking and high-speed packet classification.