TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES
Trans. Emerging Tel. Tech. (2013)

INVITED PAPER

Information-centric networking and multimedia services: present and future challenges

Giuseppe Piro[1]*, Luigi Alfredo Grieco[1], Gennaro Boggia[1] and Periklis Chatzimisios[2]

[1] DEI, Politecnico di Bari, Bari, Italy
[2] DINF, Alexander TEI of Thessaloniki, Thessaloniki, Greece

ABSTRACT

The emerging information-centric networking (ICN) paradigm is gaining worldwide the attention of academia and industrial research communities, thanks to its inherent ability to support also the next generation information and communication technology services beyond the current host-centric Internet rationale. In this context, significant scientific efforts are being devoted to the investigation of ICN-based multimedia systems because they represent the most popular use case for Internet users. In order to offer a unifying view on such valuable contributions, this paper presents a thorough review of the state of the art, a comprehensive framework to classify the most significant proposals, an overlook at the road ahead towards mature ICN-based multimedia systems and a critical discussion on available tools. Our final wish is twofold: from one hand, this paper will provide a useful starting point for beginners in the field; whereas, from the other hand, it will help researchers already working on ICN-related topics in finding a common agreement on participated scientific activities. Copyright © 2013 John Wiley & Sons, Ltd.

***Correspondence**
G. Piro, DEI, Politecnico di Bari,Via Orabona 4, 70125, Bari, Italy.
E-mail: g.piro@poliba.it

Received 7 August 2013; Accepted 22 September 2013

1. INTRODUCTION

Today, most of the Internet traffic is related to several forms of content distribution, such as file sharing, real-time applications and media streaming; users are interested to share contents rather than to interact with remote devices [1]. This 'content-centric' usage of the Internet poses new challenges that the current architecture is not able to handle in an effective manner [2]. Among them, we can mention (i) the necessity to fetch contents in a fast, reliable and effective way (i.e. availability issue); (ii) the possibility to trust contents independently from the location and the identity of who is providing them (i.e. security issue); (iii) the need to identify data no more with their locations, that is, Internet protocol (IP) addresses, but through their *names* (i.e. location-dependence issue); (iv) the seamless support of mobile users that should not experience any service interruption when moving across different access networks (i.e. mobility issue); (v) the problems related to limited storage, bandwidth and computational capabilities that affect service providers when handling a huge number of users (i.e. scalability issue); and (vi) the need to increase the resilience of information and communication technology services with respect to system failures (i.e. fault-tolerance issue) [1, 3].

All of these challenges represent the fecund *humus*, which the emerging information-centric networking (ICN) paradigm is grounding its roots on, in order to let mature new networking primitives for the Future Internet [4, 5].

But, the evolution process entailed by ICN architectures cannot underrate the forecast of data traffic trends, which all anticipate that, in the upcoming years, the Internet bandwidth will be mainly used by multimedia applications [6].

This is clearly reflected in the current ICN literature, which focuses (to a large extent) on use cases based on multimedia services [7–19].

Nevertheless, each of these valuable contributions (refer to Section 3 for an analysis on related works) only focuses on a specific issue, thus, loosing the opportunity to draw the whole picture of the road map that our community will go through in order to bring ICN multimedia services to our desktops.

To bridge this gap, the present paper provides (i) a thorough review of the state of the art; (ii) a unifying framework to classify the valuable efforts of our community;

and (iii) a critical discussion on available tools and on the future steps that will allow the ICN research to go beyond baseline scenarios.

It is important to note that, despite all ICN architectures differ in some aspects (e.g. naming scheme, data integrity, data granularity and routing strategy for both requests and data [3]), all of them assume a receiver-driven data exchange model based on content names. This actually means that all the practical guidelines related to the management of multimedia services discussed in this work, will remain valid (to a very large extent) for all ICN architectures. We also remark that among the many ICN proposals, the one designed within the named data networking (NDN) [20] project is raising a significant interest from the scientific community. Hence, because a complete and unique definition of ICN is still a work in progress at ICN research group of the Internet Research Task Force [21], we will adopt in our work, without loss of generality, the NDN terminology for the sake of clarity.

The rest of the paper is organised as follows. Sections 2 provides a synthetic background on ICN and introduces the most important issues arising from the implementation of multimedia services in this emerging paradigm. The comprehensive survey on the state of the art of architectures, protocols and algorithms enabling multimedia applications in ICN will be presented in Sections 3. Lessons learned and future challenges will be highlighted in Sections 4. Finally, Sections 5 will draw conclusions.

2. MULTIMEDIA SERVICES IN ICN

2.1. CDN and P2P versus ICN

To overcome the limitation of the classic *host-to-host* Internet model, two different solutions are currently being used to efficiently handle content distribution: content delivery networks (CDNs) and peer-to-peer (P2P) architectures.

In a CDN, contents are replicated among many servers, strategically located in different places, thus allowing users to seamlessly connect to the closest server. For this reason, many service providers (such as Akamai, Ardero, Digital Insland and Mirror Image) have experienced a massive diffusion worldwide [22, 23]. CDN architectures intrinsically offer the possibility to fix issues related to the availability and the location of contents, as well as those associated to the fault-tolerance of servers. Unfortunately, behind these advantages, it exists a non-negligible economical impact on the cost of the service either for users or content providers [24]. Moreover, despite their potentials, CDNs are not yet able to guarantee the absence of service interruptions to mobile users. In addition, in [24], it has been shown that ICN architectures can reduce the costs associated to multimedia services and can make them cheaper than CDN systems.

A cheaper, decentralised and highly scalable media distribution system can be achieved by exploiting P2P network architectures, in which all nodes (i.e. peers) may act, at the same time, as client and server [25, 26]. In these systems, peers are connected among them according to different topologies (i.e. tree, mesh and hybrid schemes), thus forming a high-level overlay network [25]. Basically, in a P2P system, a user can fetch portions of a given content, that is, chunks, from multiple sources. At the same time, he/she may share its uplink bandwidth capacity to provide chunks that have been already downloaded to other peers that request them. Even if P2P architectures are able to totally avoid issues related to the fault-tolerance of servers, they cannot offer an acceptable support to mobile users.

It is important to remark that the ICN paradigm is able to fix the aforementioned issue. On the other hand, the content distribution handled by ICN can guarantee a more efficiency and a better interoperability between services and applications [27].

2.2. Background on ICN architectures

The ICN approach is currently investigated and developed in several projects [5], such as data-oriented network architecture (DONA) [28], publish/subscribe Internet technology (PURSUIT) [29], scalable and adaptive Internet solutions (SAIL) [30], CONVERGENCE [31] and NDN [20].

The NDN and CONVERGENCE projects exploit a hierarchical ICN naming scheme where names are composed by multiple human-readable components (e.g. strings), arranged in a hierarchy. Solutions conceived within DONA, PURSUIT and SAIL projects, instead, use a flat self-certifying structure of names that supposes to identify each data through the hash of its name, eventually authenticated by using the public key of the user that have generated the considered content.

Contents are distributed within ICN architectures through two different approaches: *pull-based* and *push-based*. Pull-based schemes are adopted by DONA, NDN, SAIL and CONVERGENCE, where the user asks for a given content by issuing immediately a specific request. The push-based technique is used in the PURSUIT proposal, according to which the client interested to a specific data registers a subscription and waits to receive a notification when the requested content will be available in the future [3].

All solutions conceived so far adopt *route-by-names* strategies to forward user requests towards the closest device able to satisfy the demand (e.g. the permanent repository, namely *publisher*, or any other node that contains a valid copy in its cache) [32].

Based on such unique features, ICN architectures natively enable in-network processing of data and requests, thus allowing the optimization of service provisioning and the native support of multicast services.

2.3. NDN in a nutshell

In NDN, the communication is driven by the *consumer* of data (i.e. the user that generates the request), and only two kinds of messages are exchanged: *Interests* and *Data*. Routing operations are performed by the strategy layer only for *Interest* packets. *Data* messages, instead, just follow the reverse path towards requesting user, allowing every intermediate node to cache the forwarded content.

These activities can be executed, thanks to the exploitation of the following structures:

- The content store (CS), which is a cache memory implementing different replacement policies, such as least recently used (LRU), last frequently used (LFU) or random, where some received contents can be stored.
- The forwarding information base (FIB), which is similar to a classical IP FIB except for the possibility to have a list of faces for each content name entry; in this manner, *Interest* packets can be forwarded towards many potential sources of the required *Data*.
- The pending interest table (PIT), which is a table used to keep track of the *Interest* packets previously forwarded upstream towards content sources; it saves information about the arrival faces. In this way, backward *Data* packets can be properly delivered to the right requesters.

Each content is uniquely identified by a unique name, that is, the *Content Name*. Basically, when an *Interest* packet arrives to an NDN node, the CS is in charge of discovering whether a data item is already available or not. If so, the node may generate an answer (i.e. a *Data* packet) and send it immediately back to the requesting user. Otherwise, the PIT is consulted to retrieve if others' *Interest* packets, requiring the same content, have been already forwarded towards potential sources of the required data. In this case, the *Interest*'s arrival face is added to the PIT entry. Else, the FIB is examined to search a matching entry, indicating the list of faces which the *Interest* has to be forwarded through. At the end, if there is not any FIB entry, the *Interest* is discarded. Nevertheless, when a *Data* packet is received, the PIT table comes into play. It keeps track of all previously forwarded *Interest* packets and allows the establishment of a backward path to the node that requested the data.

In addition, other features are offered by the NDN architecture. For example, an NDN node may prohibit that its generated contents are cached, as well as a device sending the *Interest* packet could impose that the answer should be necessarily generated by the publisher, thus bypassing any intermediate caches. Also, some security features, such as the encryption and the authentication of both *Content Name* (or a part of it) and the content stored in the *Data* packet, are supported [1, 33].

2.4. Basic requirements of multimedia services in NDN

The development of ICN multimedia services requires an accurate study of all challenges associated to the management of requests, caching and routing strategies and the design of the *namespace*.

Management of content requests. According to the NDN paradigm, audio and video contents should be divided into a list of consecutive chunks that should be requested by the client during the service execution. These chunks have to be received by the client within a specific target delay (i.e. the *playout delay*) [7], which can be in the order of hundred of milliseconds or few seconds for real-time or on-demand applications, respectively. We remark that the *playout delay* is influenced not only by the network load (e.g. a network congestion may delay data delivery) but also by the request generation rate of the clients.

For instance, in order to provide seamless playback, it is important for the user to know how long to wait before assuming that an *Interest* or the corresponding *Data* packet have been lost. Knowing that, and taking into account the value of the *playout delay*, an *Interest* can be quickly reissued or it can be assumed that it is not more available (in that case, the user should move on to the next chunk of the multimedia content) [12].

On the other hand, we remark that the original receiver-driven scheme proposed for NDN, according to which a user requests one chunk at a time, could fail in the presence of real-time communications [14]. In fact, higher delays could incur because of several reasons: (i) even if the publisher generates new data, they cannot be delivered to the end user before the reception of its requests; (ii) the retransmission of *Interest* triggered by packet loss may slow down the download of the content; and (iii) routing operations may add an extra processing delay to find matches within the cache and PIT and FIB tables. Although routing operations can be made faster by adopting specific lookup mechanisms, such as those based on the exploitation of bloom filters [34], to overcome these issues, it is very important to introduce a different way for managing *Interest* packets. For example, a winning approach adopted in several works, such as [7, 14, 35] (see next section for details), is the generation of multiple *Interest* packets at the same time. In any case, it is very important to evaluate the most suitable number of *Interests* that can be kept in flight without generating congestion episodes.

Content caching and PIT tables. ICN networks can improve the performance of any streaming service by solving availability, scalability and fault-tolerance issues through the introduction of caching strategies; and thanks to features offered by the adoption of the PIT tables. From one side, data caching allows any NDN router to satisfy user requests without forwarding the *Interest* message to the remote publisher. On the other hand, the PIT table is able to limit the propagation of identical requests towards the remote server, thus offering an efficient way to redistribute contents among users (native multicast support).

Without any doubts, the design of advanced caching strategies could notably improve the performances of on-demand multimedia streaming applications [8, 15]. Nevertheless, some recent works demonstrated that content caching is not a mandatory goal in the presence of real-time services [7, 14] because cached contents may expire before being asked in a second time.

Routing strategies. Particular attention should be devoted to the routing strategy, which is in charge of forwarding user requests towards those nodes able to provide the asked content. The routing algorithm influences several aspects in an ICN architecture, such as network latency, in-node and network scalability, delays related to the consultation of routing tables and so on [32]. From the point of view of multimedia applications, the impact that the routing algorithm has on network latency could significantly impair the quality of the service, especially for real-time communications. High latency, in fact, may lead to a frequent violation of the *playout delay*, with a consequent decrement of service quality. In this context, as already remarked in [32], the choice of the *namespace* could be one of the critical aspects influencing the behaviour of the routing strategy. As a consequence, the design of an effective routing algorithm should be jointly accomplished with the definition of *names*.

***Namespace* design.** In a typical IP architecture, the initialization of a multimedia session is accomplished; thanks to the knowledge of the IP address and the port of remote devices. In ICN, it is required to map the service initialization to a content-oriented model; and, in this regard, the design of the *namespace* plays a key role.

The *name tree* could be created by adopting the *constructible name* concept, firstly introduced in [35], according to which users should share a scheme and/or an algorithm to create *names* identifying, in a unequivocally manner, remote device and contents it generates.

Furthermore, we note that interactive applications, such as audio and video calls, require the generation of a bidirectional flow enabling users to exchange data during the time [19]. Let us suppose to consider a given client that contacts a remote user to initialize a media conversation. In conventional IP networks, the remote user is able to identify the client by extracting some information from IP and User Datagram Protocol (UDP), or Transmission Control Protocol (TCP), headers. Hence, it is immediately able to establish with this user a parallel and opposite connection. In NDN, instead, the *namespace* should be designed to include details within the *Content Name* of the first generated *Interest*. In this way, in fact, the remote device will be aware of the *name* associated to the client, which is used to configure an opposite connection by simply extracting information from the received request.

3. MULTIMEDIA SERVICES IN ICN: STATE OF THE ART

In this section, we provide an accurate description of the main proposals presented in literature, that is, [7, 8, 10–19, 35], classifying them according to the considered multimedia service, the presence of a dedicated *namespace* hierarchical structure, the addition of novel functionalities to the basic NDN behaviour and the performance evaluation methodology (a summary of our findings has been reported in Table I).

3.1. ICN architectures for voice services

3.1.1. Voice Over Content-Centric Networks.

The possibility to map existing voice over IP (VoIP) services into NDN, in a way which preserves security, interoperability and good service performance, has been firstly discussed in [35], where it has been proposed the voice over content-centric networks (VoCCN) architecture.

Main features covered by VoCCN are (i) the initialization of the communication through the *service rendezvous* procedure; (ii) the request and *on-demand publishing* of contents that have not been yet generated before the initialization of the call; (iii) the generation of a bidirectional flow between users; (iv) the creation of a *Content Name* without knowing the content *a priori* (i.e. *constructible name*); (v) the multiple *Interests* generation; (vi) the support for both session initiation protocol (SIP) [36] and real-time transport protocol (RTP) [37] payloads, and (vii) the interoperability with standard-conforming VoIP implementations.

First of all, a deterministic algorithm is used to generate *names* for identifying both users and contents in any phase of service execution (i.e. initialization, creation of bidirectional flows and exchange of media contents).

The *service rendezvous* procedure is exploited to configure the VoIP session through the SIP protocol. Each user, identified by a unique $userId$, announces its availability for the voice service by registering itself to the *namespace*:

/domain/sip/userId/invite.

The user wishing to initialize a call (i.e. the caller), maps a SIP INVITE message to an *Interest* packet that will be forwarded to the remote user (i.e. the callee). The *Content Name* of the first *Interest* packet is built by appending to the aforementioned *namespace*—the content of the SIP INVITE message. The caller may also decide to encrypt the message using a randomly generated symmetric key, sk, that will be encrypted with the public key* of the callee, pkB. The resulting *Content Name* of the *Interest* packet will be:

$$/domain/sip/calleeId/invite/E_{pkB}(sk)/E_{sk}(InviteMessage)$$

The callee answers to this request by generating a *Data* packet containing the SIP response. From this moment on, the exchange of media contents is performed using the RTP.

Information exchanged during the rendezvous procedure are used to construct *names* that identify, in the VoCCN *namespace*, each portion of the media stream. To each

---

*Public keys of users can be distributed within the network via NDN.

Table I. Summary of proposals presented in literature.

	Target multimedia service			Extensions to the NDN architecture			Performance evaluation approach		Target technology		
Work	Voice over Internet protocol	Video on demand	Video real-time	CCNx messages	Named data networking router functionalities	New *Interest* management	Real experiments	Simulations	Wireless	Wired	Ad hoc design of the *namespace*
Voice over content-centric networking [35]	x						x			x	x
Audio conference tool [19]	x						x			x	x
Time-shifted TV services [15]		x			x			x		x	
Content-centric networking live streaming [18]		x					x			x	
Time-based Interest protocol [17]			x	x	x	x	x			x	
MERTS [14]			x	x	x	x	x			x	
Content-centric networking-TV [7]			x	x	x	x		x		x	
Named data networking in heterogeneous wireless test-bed [8]		x					x		x		
Adaptive retransmission scheme[11]			x			x		x	x		
Dynamic adaptive streaming over HTTP-based architecture[10]		x					x		x		x
Named data networking video[12]		x	x							x	x

media chunk is assigned an unique sequence number and it is identified by the *Content Name*:

/domain/userId/call-id/rtp/seq-no

To solve the problem related to network latency, the caller can generate and send multiple *Interest* packets at the same time. Every time a new *Data* packet is received, a new *Interest* is released, thereby restoring the total number of pending *Interests*.

Finally, a stateless VoCCN-VoIP gateway has been designed in order to achieve interoperability between VoIP and VoCCN. It translates VoIP packets (SIP and RTP) to VoCCN packets (which again merely encapsulate SIP and RTP messages) and vice versa.

To demonstrate the effectiveness of VoCCN, a prototype architecture has been tested under real machines, implementing an extension of the open source Linux VoIP phone, that is, Linphone (version 3.0). Experiments have shown that VoCCN is able to guarantee functionally and performance equivalent to traditional VoIP applications, even if adopting an network architecture that is simple to implement and easy to configure.

3.1.2. Audio Conference Tool.

Despite its interesting features, VoCCN only supports a voice communication between two users. A more complex architecture enabling audio conference services, called audio conference tool (ACT), is described in [19].

Differently from IP-based systems that relay on a centralized node for the conference management, ACT exploits the named data approach to discover ongoing conferences, as well as speakers in each conference, and to fetch voice data from individual speakers. The developed architecture is completely distributed and robust against component failures.

Similarly to the previous contribution, a specific *namespace* and an algorithm to create *names* have been tailored to support all these tasks.

The conference discovery procedure is used to retrieve the list of ongoing conferences and to announce a new conference to potential participants.

A conference initiator announces the conference by creating a data object providing all details about that event in the session description protocol format [38]. These data are stored in the application buffer until the conference ends.

When a user wants to participate to a conference, it should learn from the network the list of active ones. To this end, he/she floods an *Interest* packet with *Content Name* set to

/ndn/broadcast/conference/conference-list.

A node storing SDP/session description protocol data objects in their repository/cache will be able to provide an answer to that request by sending the corresponding *Data* packet.

Participants of a conference fall in two categories: speakers and listeners. The former group of users produces voice data, whereas the latter one is only interested to listen to the speech. Basing on this assumption, in the ACT architecture, a user only needs to know the list of speakers in a conference in order to retrieve the data from them. Hence, when it joins a conference, he/she will discover active speakers by issuing an *Interest* packets with the *Content Name* set to

/ndn/broadcast/conference/conference-name/speaker-list.

Also in this case, any nodes in possession of such information should be able to answer to the user request by generating the corresponding *Data* packet.

During the audio conference, each speaker stores generated data packet in a circular buffer. A *Data* packet is identified by a unique segment number, which is appended to the end of the *Content Name*, as reported in the succeeding texts:

/ndn/conference/conference-name/speaker-name/seg-num.

If *seq-num* is not explicitly specified, ACT will assume that the *Interest* message is asking for the most recent *Data* packet.

Following the same approach adopted in [35], the impact of high network latency are counteracted by sending multiple *Interest* messages at the same time. Whenever a new *Data* packet comes back, the user will issue a new *Interest*.

A proof-of-concept ACT architecture has been implemented within the open source Mumble project [39] by using CCNx primitives [40]. Tests demonstrated that ACT may offer a quite good subjective voice quality.

3.2. ICN architectures for video on-demand services

3.2.1. CCN Live Streaming Architecture.

The NDN live streaming architecture has been designed to enable simple HTTP Live Streaming services in an NDN network [18]. The remote server breaks the video stream into a sequence of segments (i.e. video chunks) and stores them within a local repository. Moreover, in order to describe the structure and the properties of that media content, it generates a proper index file, called playlist. The playlist is the first data that the client has to retrieve before starting to the download of the video.

During the download of video chunks, HTTP Live Streaming requests, that are generated by the media player, are translated by the HTTP proxy, which is hosted at the client side, into *Interest* packets. These requests will be delivered in the NDN architecture towards the remote device that is able to provide the requested data. When the user receives a *Data* packet, the HTTP proxy converts such a message into an HTTP response, which will be delivered to the media player that handles the video reproduction.

The proposed solution has been implemented in a real testbed, and experiments have demonstrated that the adoption of the NDN architecture is able to increase the download speed and to reduce the download delay when the overall number of users, interested in a given content, increases.

3.2.2. Cooperative caching strategy enabling time-shifted TV services.

The work presented in [15] introduces a cooperative caching strategy for large video streams in on-demand services. Authors focus the attention on *time-shifted TV* services that allow viewers to watch their favorite broadcast TV programs within an expanded time window.

The main goal of the solution conceived in [15] is the reduction of the number of *Interest* packets delivered to the remote server.

It is supposed that the i-th NDN router is identified by a *label*, l_i, computed during the *initialization stage procedure* through the execution of a specific distributed algorithm.

Let c_j be the identifier (ID) of the j-th chunk of the content c. Now, the possibility to cache the j-th chunk is given to the i-th NDN node only if

$$c_j \; mod \; k = l_i$$

The cooperative caching strategy is implemented through the adoption of two new tables, that is, the *collaborative router table*, storing the label of neighbor routers, and the *collaborative CS* (CCS) table, storing contents that could be cached by other nodes in the neighborhood.

When an NDN node receives an *Interest* packet, it verifies if the *Content Name* matches at least one entry into CS, CCS, PIT and FIB (in the specified order). This will enable the router to generate the *Data* packet or to properly select the node which the request is forwarded to.

When a *Data* packet storing the chunk c_j is received and a PIT match is found, the i-th router will forward such a message to the interface indicated by the PIT table. Then, it will compute

$$c_j \; mod \; k$$

in order to decide the fate of the packet.

If $c \; mod \; k = l_i$, the data will be cached. Otherwise, the router will send the chunk to the neighbor device stored within the collaborative router table and having the right label; it will add the received chunk ID in the CCS Table.

The effectiveness of the proposed scheme has been investigated through computer simulations conducted with an owner tool developed in the OMNET++ environment. Results demonstrated that the cooperative caching strategy is able to reduce the number of requests forwarded to remote broadcast TV servers, thus guaranteeing, at the same time, a load balancing among them. On the other hand, the average response time registered by the proposed algorithm is a bit higher with respect to the one obtained with the adoption of the LRU scheme. However, authors remarked that this result does not lead to a significant degradation of the quality of service offered to end users.

3.3. NDN in a heterogeneous wireless test-bed

A recent research paper [8] evaluated the performances of video on-demand streaming applications handled through the NDN paradigm in a heterogeneous wireless network. The investigated scenario is composed by a group of users that are interested in watching, at the same time, a given video content available at the remote streaming server. Each device implements the CCNx protocol stack [40], and it is equipped with two different network interfaces: the cellular interface providing the Internet connectivity and the WiFi interface used to create a local and ad hoc wireless network. Results demonstrated that, in this case, the ICN approach is able to significantly offload the cellular network without compromising the quality of service experienced by end users. In fact, segments of the video content are downloaded through the cellular interface by only one user and then shared inside the ad hoc network by using NDN.

3.3.1. DASH-based architectures.

The adaptive mobile video streaming in wireless NDN (AMVS-NDN) has been proposed in [10]. It has been designed to offer video streaming services through the emerging dynamic adaptive streaming over http (DASH) scheme [41].

According to DASH specifications, the video is divided in chunks of fixed or variable length. For each chunk, several segments, each one encoded with a specific frame rate, are available at the server side. The list of chunks, segments, frame rates and other parameters associated to the video are organized according to a so called media presentation description (MPD) file [42]. Hence, in order to identify multiple copies of the same video content, each one having a specific set of properties, a specific *namespace* have been designed.

Adaptive mobile video streaming in wireless-NDN has been evaluated in a heterogeneous wireless environment in which a number of devices are equipped with both 3G and WiFi network interfaces. The former one provides the Internet connectivity, whereas the latter one can be used to form an ad hoc wireless network.

A user that wants to download a video content, sends the *Interest* packet to retrieve the corresponding MPD file. The adopted *Content Name* is

/domain/fileName/_INIT.

Once the MPD file has been obtained, the user can start the download of the video content by requesting segments encoded with the lowest bit rate (conservative approach).

Each video segment is uniquely identified by the following *Content Name*:

/domain/fileName/videoQuality/SegNum.

During the reception of the video, the client can estimate the available bandwidth and dynamically adapt the video encoding rate (and hence its quality).

At the strategy layer (where routing operations are implemented), each device tries to fetch the video segment sending the *Interest* packet over the WiFi network; if no answers will be received, the same *Interest* will be forwarded to the cellular network interface. The received data will be stored in the local cache and will then be shared with other devices in the ad hoc network by means of NDN primitives.

Computer simulations highlighted the performance gains achievable with the adoption of the presented scheme if compared with respect to the basic DASH architecture (i.e. without the ICN approach).

The performance of DASH in NDN has been evaluated also in other recent papers, such as [13, 16, 43], finding similar results to the ones reported in [10].

3.4. ICN architectures for real-time video services

3.4.1. The MERTS platform.

The first architecture enabling real-time services that we consider in our discussion is the MERTS platform, which has been designed to handle, at the same time, real-time and non-real-time flows in NDN [14]. It aims at (i) classifying the type of service (TOS) to which each packet, for example, *Interest* and *Data* messages, belongs to when it passes through a given NDN node; (ii) limiting the impact of network latency by issuing multiple requests at the same time; and (iii) adapting the caching policy to the TOS associated to the received *Data* packet.

The first goal is reached by extending the basic structure of the *Interest* packet by using the TOS field to differentiate real-time and non-real-time traffics. Then, in order to serve real-time applications, the *one-request-n-packets* strategy is proposed, according to which the user issues a *Special Interest* (SI) asking for *n* consecutive *Data* packets. When all the *n* chunks will be received by the user, a new SI is generated.

The right behavior of the proposed scheme can be guaranteed only by ensuring that the PIT entry associated to a SI will not be deleted after the reception of only a subset of chunks requested with the SI itself. To this aim, the normal functionality of the PIT table is modified by imposing that the SI can be erased only after the expiration of its *life time*.

Authors remarked that the content caching is not really needed for real-time flows. For this reason, they propose to disable the caching of multimedia *Data* packets. In this way, several advantages are reached: firstly, the memory utilization of the cache is optimized, and the performances of non real-time services are notably improved; secondly, this approach reduces the delay provoked by the search of the *Data* in the cache, thus limiting the overall network latency.

The compatibility with the common NDN architecture is managed by the introduction of a flexible transport mode selection scheme; knowing the TOS associated to an *Interest* or *Data* packet, each NDN node can adapt the set of operations to do against it (e.g. delete or not delete a SI from the PIT, as well as cache or not cache the received data).

Real experiments have demonstrated that the proposed solution is able to reduce the total number of *Interest* packets handled by the network and the average downloading time, at the cost of a slight increment of the overall packet loss ratio.

3.4.2. The CCN-TV architecture.

Real-time TV services can be offered with a recently proposed CCN-TV architecture [7, 44]. Main aspects covered in CCN-TV are (i) the definition of a *channel bootstrap phase*; (ii) the design of an effective flow control strategy; and (iii) the adoption of an efficient mechanism for retransmitting *Interest* packets. All of these functionalities can be enabled by extending the basic structure of the *Interest* packet by means of an additional *Status* field, which is in charge of recognizing whenever this message is related to the channel bootstrap phase or to a retransmission.

During the *channel bootstrap phase*, the client bootstraps a TV channel by releasing an *Interest* packet with the *Status* field set to BOOTSTRAP, thus imposing every intermediate NDN node to enforce the forwarding of this request to the remote server, triggering the generation of *Data* packet containing the last useful generated video frame. Once the user received this *Data* packet, it will request subsequent chunks using a sliding window mechanism. Firstly, let us define *Pending Chunk* and *Pending Window* as the chunk whose *Interest* has been sent by the node and the window containing *W* different pending chunks not yet received, respectively. In details, together with the *chunkID*, we store in the *Pending Window* the timestamp of the first request and the timestamp of the last retransmission. Hence, whenever new data are received, or if the node does not receive any data for at least *windowTimeout* seconds, the following operations are performed: (i) purge the *Pending Window* from all the chunks expired that have already been played; (ii) retransmit all chunks that have not been received within the *windowTimeout*; and (iii) transmit, for each slot that got freed by the received or expired chunks, the *Interest* for a new one. Normally, an NDN node does not propagate *Interest* packets related to contents already requested by other users in the past but not yet satisfied with the corresponding *Data* packets. To force the propagation of retransmitted *Interests*, the *Status* field is set to RETRANSMISSION, and each intermediate NDN router is forced to propagate it versus the node that can satisfy this request (i.e. by skipping the usual NDN mechanism).

Computer simulations conducted by using *ccn-tv-sim* [45] have shown the effectiveness of the discussed architecture. Moreover, presented results have highlighted that, differently from any caching policies, the PIT has a fundamental role in reducing the load at the server side in the presence of real-time streaming services.

3.4.3. Adaptive retransmission scheme for real-video streaming.

The idea to retransmit a request not yet satisfied after a given timeout has been considered also in [11], where authors showed that the best timeout value is very hard to predict. Well-known approaches, such as the one used by the TCP for estimating the retransmission timeout (RTO), could fail in this task because they do not consider the round-trip time (RTT) variation caused by in-network caching. On the other hand, in a wireless network, there is a high probability to interpret the loss of a packet because of the channel error as a congestion episode. As a consequence, it is difficult to adapt the generation rate of *Interest* packets.

Starting from these observations, the work [11] proposes a novel and efficient retransmission scheme of *Interest* packets, which has been conceived to reduce video packet losses in an NDN network.

From one side, the lifetime estimation algorithm captures the RTT variation caused by the in-network caching and dynamically evaluates the value of the RTO. From another hand, the explicit congestion notification field is added to the *Data* packet for signaling the ongoing network congestion to the end user. This information will be used by the retransmission control scheme to differentiate channel errors from network congestion episodes. Hence, based on the reason of packet losses, this algorithm adaptively adjusts the retransmission window size, that is, the number of total *Interests* that can be retransmitted by the client.

A simulation study, executed with an owner ns-2 module, reported that the devised approach is able to improve the quality of video stream experienced by end users.

3.4.4. Time-based Interest protocol enabling real-time streaming services in CCN.

Another efficient scheme enabling real-time streaming services in an NDN network has been presented in [17]. This work remarks that the sending of multiple *Interest* packets at the same time (that is the approach adopted in some of aforementioned papers, such as [14] and [7]) may cause both processing overheads and waste of the uplink bandwidth. This issue is resolved by introducing a new *Interest* packet, sent by the user to ask for a group of contents generated by the publisher during a specific time interval. During such time interval, all chunks generated by the remote server or transferred from other nodes can be delivered to the user. We note that also this scheme requires the modification of the normal behavior of an NDN router: the *Interest* packet should not be deleted from the PIT until its *life time* expires. Real experiments have demonstrated that the proposed approach is able to reduce the delay jitter and is able to increase the aggregate throughput achieved by video flows.

3.4.5. NDN Video architecture.

The final architecture that we consider in our study is NDNVideo, which has been designed and implemented on top of CCNx, in order to offer both real-time and non-real-time video streaming services [12]. A first important issue addressed by the NDNVideo project is the design of the *namespace* that enables the publisher to uniquely identify every chunk of the multimedia content and allows the consumer to easily seek a specific place in the stream. In particular, the *Content Name* is built in order to provide information about the video content, the encoding algorithm and the sequence number associated to a given chunk:

/domain/fileName/codecInfo/SegNum.

Nevertheless, to provide a random-access to the video content during the seeking procedure, the user can specify, within the *Content Name*, a timecode (e.g. HH:MM:SS:FF) that will be used by the server to select the most suitable *Data* packet within the video stream:

/domain/fileName/codecInfo/HH:MM:SS:FF.

After this request, the user will ask for video data using consecutive segment numbers.

To support real-time streaming services, the client may issues multiple *Interest* packets at the same time. However, to avoid that it will fetch data too quickly and request segments that do not yet exist, the client estimates the generation rate of *Interest* packets by knowing the time at which previous data packets were generated by the publisher (this information is stored in a specific field of the *Data* packet). Each unsatisfied *Interest* packet can be retransmitted during a time window whose duration is equal to the playout delay. Finally, a low-pass filter similar to the one defined in the TCP is adopted to adjust the RTO based on previous RTT values.

4. LESSONS LEARNED AND FUTURE CHALLENGES

We would like to conclude our work by presenting a list of lessons learned during the study of the state of the art and then by giving some details about the uncovered issues that need to be considered by researchers in the near future. Moreover, also the list of research instruments that can be exploited to carry out research activities is provided.

4.1. Lessons learned

Tables II and III report the summary of features, techniques and novel approaches proposed in literature. Accordingly, we can conclude the following:

Table II. *namespace* definition, *Interest* management and messages extensions proposed in literature.

Work	*namespace* definition	management of *Interest* packets	modification of CCNx messages
Voice over content-centric networking [35]	Control messages and user details are stored in the *Content Name*; media chunks are identified by an unique sequence number	Generation of multiple contemporary requests	None
Audio conference tool [19]	Specific *names* are used to discover conferences and speakers; media chunks are identified by a unique sequence number	Generation of multiple contemporary requests	None
Time-shifted TV services[15]	Not addressed	as in NDN	None
Content-centric network live streaming [18]	Not addressed	as in NDN	None
Time-based interest protocol [17]	Not addressed	an *Interest* asks for data generated within a specific time interval	None
MERTS [14]	Not addressed	an *Interest* asks for *n* consecutive data	the type of service field is added to differentiate flows
Content-centric network-TV [7]	Not addressed	generation of multiple contemporary requests; *Interest* retransmissions handled through a sliding window mechanism	the *Status* filed has been introduce for detailing the type of the *Interest* packet
Named date networking in heterogeneous wireless test-bed [8]	Not addressed	*the same of NDN*	None
Adaptive retransmission scheme [11]	Not addressed	generation of *Interests* handled through the estimation of the timeout add the congestion detection	the explicit congestion notification field in the *Data* packet announces the network congestion
Dynamic adaptive streaming over HTTP-based architecture [10]	a special *Interest* is defined to discover the MDP file; media chunks are identified by a unique sequence number	*as in NDN*	None
Named data networking video [12]	media chunks are identified by a unique sequence number; the timecode is added to the *name* to seek the video stream	generation of multiple contemporary requests performed by knowing the timeout and the *playout delay*	None

NDN, named data networking.

Table III. New functionalities proposed in literature for named data networking devices.

Work	Novel functionalities of a NDN node	Novel approaches adopted by the strategy layer
Voice over content-centric networking [35]	None	None
ACT [19]	None	None
TIme-shifted TV services [15]	Cooperative caching strategy with the introduction of CTR and Collaborative content store tables	*Interests* are forwarded according to information stored in both collaborative router table and collaborative Content Name tables
Content-centric networking live streaming [18]	The client uses the HTTP proxy to map HTTP Live streaming packets in NDN messages, and vice versa	None
Time-based Interest protocol [17]	*Interests* are delated only after their expiration	None
MERTS [14]	*Interests* are delated only after their expiration; the cache is disabled for multimedia contents	None
Content-centric networking-TV [7]	Possibility to disable the cache for multimedia contents BOOTSTRAP and RETRANSMITTED *Interests* are always forwarded to the remote server	
NDN in heterogeneous wireless test-bed [8]	None	*Interests* are preferably forwarded through the WiFi connection instead of to the cellular one
Adaptive retransmission scheme [11]	None	None
Dynamic adaptive streaming over HTTP-based architecture [10]	None	*Interests* are preferably forwarded through the WiFi connection instead of to the cellular one
Named date networking video[12]	None	None

- The design of the *namespace* is a critical issue for the implementation of multimedia services.
- Clients and publishers should know *a priori* how creating *names* according to the defined structure of the *namespace*, thus being able to contact a remote user without knowing its location and to uniquely identify portions of video and audio contents.
- The generation of contemporary multiple requests could limit the negative impact on network latency.
- The adoption of a sliding window mechanism to control the transmission (and retransmission) of *Interest* packet based on the knowledge of the timeout and the *playout delay* represents a valid technical solution, but it needs to be properly designed in order to avoid the waste of bandwidth and an excessive exploitation of computational resources.
- The role of the cache is mandatory only for on-demand services.
- Cooperative caching strategies may improve the behaviour of the network.
- A specific routing strategy may optimize the resource utilization in heterogeneous networks.

At the time of this writing, we believe that our findings could be of high aid to all the people interested in this research field, who would like to design and improve ICN-based architecture enabling multimedia services. In fact, thanks to the definition of basic and significant guidelines, these lessons learned can pave the way towards the development of more complex and complete solutions.

4.2. Future challenges

We are aware that, despite the high number of works already presented in literature, there are a lot of issues that need to be accurately investigated before really deploying multimedia services in ICN architectures.

First of all, security aspects merit more attention. Since from its original definition, NDN intrinsically supports security capabilities (i.e. encryption and authentication of both *Interest* and *Data* packets), which can be directly adopted also for multimedia services. In line with [33], security materials (e.g. the public key of the publisher) can be exchanged among nodes by using the fundamental mechanism offered by NDN: an *Interest* packet is released to ask for a public key and a *Data* packet, containing that key, is generated and delivered to the requester. Nevertheless, we can imagine a more complex scenario with a security authority that will sign public keys of users, thus guaranteeing their authenticity. Moreover, users may exploit specific schemes, always based on the exchange of *Interest* and *Data* packets, to negotiate ephemeral session keys for protecting their data. None of these aspects have been already addressed, and for this reason, the definition of secure multimedia architectures is one of the most important challenges that has to be addressed in the near future. Moreover, we believe that it could be also useful to study the impact that such solutions may have on both computational and bandwidth overheads.

Most of the proposals highlighted the importance of cache may have on system performance, especially for on-demand streaming services. A key consideration is that it is impossible to consider intermediate routers with infinite storage capabilities. As a consequence, the limited amount of resources in each cache could affect the behavior of multimedia services. We note that this thought could become more evident in heterogeneous scenarios where network devices have to handle different flows (not only belonging to multimedia applications) with different requirements. Thanks to the study conducted in [14] and [7], we can design an ICN platform where it is avoided to cache contents belonging to real-time communications. This will certainly release a lot of memory to other kinds of flows. On the other hand, more sophisticated caching strategy, for example, based on cooperative approaches as suggested in [15], may additionally optimise the utilization of network resources. Finally, we cannot forget that the size of a cache has its negative impact on the economical point of view. The higher is the cache, the higher is its cost and the computational capabilities required to handle the CS and retrieval. Based on these considerations, it is evident that study on caching related issues must be widely faced in the future. This should allow a better sizing of the entire network architecture and the estimation of the quality of deployed multimedia services.

Similar considerations can be performed also for the size of the PIT table. We showed that the PIT table could reduce the amount of requests delivered to remote servers. Unfortunately, like for the cache, we cannot store infinite entries within this data structure. The generation of multiple *Interest* packets, which represents the most common approach adopted for real-time applications, could lead to the saturation of PIT tables within intermediate routers. This effect could be more deleterious when the timeout variable, which influences the retransmission rate of not satisfied requests, is not correctly dimensioned. To fix these problems, preliminary schemes has been proposed in [17] and [14], where special *Interest* packets requesting for multiple data chunks have been devised. However, such approaches need to leave the *Interest* until the expiration of its *life time*, and in high-loaded scenarios where several users are interested to the same content, they may provoke an excessive waste of network bandwidth because same data can be delivered multiple times to end users. To provide an answer to all of these issues, more deep studies must be carried out in the future.

As remarked in Section 2, the routing strategy is one of the most critical issue for ICN systems. Nevertheless, a part of very simple approaches proposed in [8] and [10], no further sophisticated techniques have been envisaged so far. The design of optimised routing strategies in ICN platforms, in the presence of multimedia services, is a totally uncharted topic that can attract most of the researchers working in this field.

To conclude, other important open issues are related to the design of congestion control mechanisms. At the time of this writing, some strategies, which exploits window-based additive increase multiplicative decrease approaches, have been proposed in literature (see e.g. [27] and [46]). However, the behavior of these solutions has not been fully studied, and the impact that they have on multimedia streaming services has not been investigated yet.

4.3. Available research instruments

Fortunately, there are many instruments that can support any kind of research activities devoted to the design of solutions aimed at solving all the envisaged future challenges. Several simulation frameworks, software and real testbeds can be exploited to this purpose.

Among all the available simulation platforms, *ndnSim* [47] represents, nowadays, the most complete and reliable one. It has been developed on top of the NS-3 simulation framework and implements communication models devised within the NDN project, including all data structures (i.e. CS, PIT and FIB), application messages, network entities (i.e. routers and faces), caching policies (i.e. First In First Out (FIFO) and LRU) and a forwarding strategy based on the DijkstraÕs shortest path algorithm. Another diffused simulation tool is *ccnSim*, that is, an open source and scalable chunk-level simulator of NDN built on top of the OMNET++ framework [48]. Also, *ccnSim* implements all the main aspects of an NDN architecture. However, developed network models are very simplified and just offer NDN functionalities from the system level point of view. The latest emerging simulation tool based on *ccnSim* is *ccn-tv-sim* [45], which has been developed to support research activities presented in [7].

Both *ndnSim* and *ccnSim* do not support, as default, the implementation of multimedia applications. Their upgrades are hence required before using them to study the transmission of video and voice contents in an NDN scenario. Instead, *ccn-tv-sim* implements video applications whose video flows have been obtained from real clips, encoded with a number of encoding rates in the range [200, 1000] kbps. In addition, it also offers the implementation of the sliding window mechanism for the *Interests* management, as described in [7].

The adoption of a simulator is the first step to evaluate the performance of novel ideas. But very often, because of the approximation level that characterizes all of these instruments, this approach could detach the research activity from the reality. To bridge this gap, it is necessary to use a software able to test protocols and algorithms in real testbeds.

The official implementation of the NDN communication paradigm is *CCNx* [40]. It can be also used on top of the NS-3 simulator thanks to the NS3 DCE CCNx module [49]. In addition, also a lightweight implementation of CCNx exists, that is, *CCN-lite* [50], useful when there are computational capability constraints in devices exploited for carrying out experiments. Another recently proposed tool is *CCN-Joker*, which offers an application-level overlay, based on the NDN paradigm, that can be applied to both emulation-based analyses and real experiments [51]. In addition, we remark that the NDNVideo platform, which already implements several features to support video streaming applications on top of the CCNx protocol stack, can be already exploited to design, test, extend and optimize any other mechanism enabling ICN multimedia services. We note that, differently from all the presented simulation tools, *CCNx*, *CCN-Joker* and NDNVideo could interact with real applications, thus making the study as realistic as possible. On the other hand, they can be also used in Planetlab, Grid5000 and Onelab platforms in order to perform large-scale experiments.

5. CONCLUSIONS

In this paper, we investigated the main issues related to the development of multimedia applications in ICN architectures. The accurate literature review allowed us to provide a unifying framework for classifying all the proposals conceived so far, as well as the definition of few, but very significant, lessons learned on this topic. In particular, our study highlighted the main guidelines to design the namespace, the management of user requests and the implementation of both caching and routing strategies in ICN-based multimedia systems. Moreover, the challenges worth of investigation have been presented along with currently available tools, in order to foster a common ground of understanding among researchers working on ICN-related topics.

ACKNOWLEDGEMENTS

This work was supported by the PON projects (RES NOVAE, ERMES- 01-03113, DSS-01-02499 and EURO6-01-02238) funded by the Italian MIUR and by the European Union (European Social Fund) and Greek national funds ESPA 2007-2013, EDULLL Archimedes III.

REFERENCES

1. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL. Networking namedcontent, In *ACM CoNEXT '09*, Rome, Italy, 2009; 1–21.
2. Ahlgren B, Aranda PA, Chemouil P, Oueslati S, Correia LM, Karl H, Sollner M, Welin A. Content, connectivity, and cloud: ingredients for the network of the future. *IEEE Communications Magazine* July 2011; **49**(7).
3. Ahlgren B, Dannewitz C, Imbrenda C, Kutscher D, Ohlman B. A survey of information-centric networking. *Communications Magazine, IEEE* July 2012; **50**(7): 26–36.
4. Matsubara et al. D. Toward future networks: a viewpoint from ITU-T. *IEEE Communications Magazine* 2013; **51**(3): 112–118.

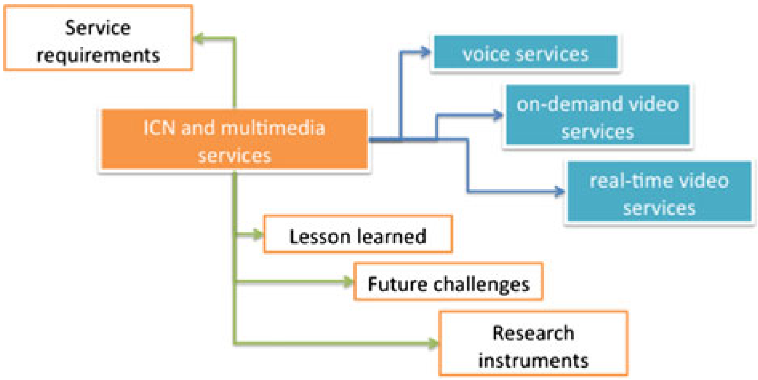
5. Xylomenos G, Ververidis C, Siris V, Fotiou N, Tsilopoulos C, Vasilakos X, Katsaros K, Polyzos G. A survey of information-centric networking research. *IEEE Communications on Surveys & Tutorials* 2013; **PP**(99): 1–26.
6. Cisco visual networking index: Forecast and methodology, 20122017, February 2013. White Paper.
7. Ciancaglini V, Piro G, Loti R, Grieco LA, Liguori L. CCN-TV: a data-centric approach to real-time video services, In *Proceedings of IEEE International Conference on Advanced Information Networking and Applications, AINA*, Barcelona, Spain, 2013; 982–989.
8. Detti A, Pomposini M, Blefari-Melazzi N, Salsano S, Bragagnini A. Offloading cellular networks with information-centric networking: the case of video streaming, In *Proceedings of IEEE Int. Conf. on a World of Wireless, Mobile and Multimedia Networks, WoWMoM*, San Francisco, CA, 2012; 1–3.
9. Grieco LA. Emerging topics: special issue on multimedia services in information centric networks (guest editorial). *IEEE COMSOC MMTC E-letter* 2013; **8**(4): 4–5.
10. Han B, Wang X, Taekyoung Kwon NC, Choi Y. AMVS-NDN: Adaptive mobile video streaming and sharing in wireless named data networking, In *Proceedings of IEEE Int. Workshop on Emerging Design Choices in Name-Oriented Networking, NOMEN*, Tourin, Italy, 2013.
11. Han L, Kang S, Kim H, In H. Adaptive retransmission scheme for video streaming over content-centric wireless networks. *IEEE Communications Letters* 2013; **PP**(99): 1–4.
12. Kulinsky D, Burke J, Zhang L. Video streaming over named data networking. *IEEE COMSOC MMTC E-letter* 2013; **8**(4): 6–9.
13. Lederer S, Mller C, Rainer B, Timmerer C, Hellwagner H. Adaptive streaming over content centric networks in mobile networks using multiple links, In *Proceedings of IEEE Int. Conf. on Communication, ICC*, Budapest, Hungary, June 2013.
14. Li H, Li Y, Lin T, Zhao Z, Tang H, Zhang X. MERTS: a more efficient real-time traffic support scheme for content centric networking, In *Proceedings in IEEE Int. Conf. on Computer Sciences and Convergence Information Technology, ICCIT*, Dhaka, Bangladesh, 2011; 528–533.
15. Li Z, Simon G. Time-shifted TV in content centric networks:the case for cooperative in-network caching, In *Proceedings of IEEE ICC*, Kyoto, Japan, 2011; 1–6.
16. Liu Y, Geurts J, Point JC, Lederer S, Rainer B, Mueller C, Timmerer C, Hellwagner H. Dynamic adaptive streaming over CCN: a caching and overhead analysis, In *Proceedings of IEEE Int. Conf. on Communication, ICC*, Budapest, Hungary, June 2013.
17. Park J, Kim J, Jang MW, Lee BJ. Time-based interest protocol for real-time content streaming in content-centric networking (CCN), In *Proceedings of IEEE Int. Conf. on Consumer Electronics, ICCE*, Barcelona, Spain, 2013; 512–513.
18. Xu H, Chen Z, Chen R, Cao J. Live streaming with content centric networking, In *Proceedings 3rd Int. Conf. on Networking and Distributed Computing*, Hangzhou, China, 2012; 1–5.
19. Zhu Z, Wang S, Yang X, Jacobson V, Zhang L. ACT: audio conference tool over named data networking. In *Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking*. ACM: New York, NY, USA, 2011; 68–73.
20. Zhang L, Estrin D, Burke J, Jacobson V, Thornot JD, Smatters DK, Zhang B, Tsudik G, Krioukov D, Massey D, Papadopulos C, Abdelzaher T, Wang L, Crowley P, Yeh E. Named data networking (NDN) project. *PARC Technical Report TR-2010-02*, 2010.
21. Pentikousis K, Ohlman B, Corujo D, Boggia G, Tyson G, Davies E, Mahadevan P, Spirou S, Molinaro A, Gellert D, et al. *ICN Baseline Scenarios and Evaluation Methodology, draft-pentikousis-icn-scenarios-04*. IETF ICNRG working group, July 2013. Available from: http://tools.ietf.org/html/draft-pentikousis-icn-scenarios-04.
22. Pallis G, Vakali A. Insight and perspectives for content delivery networks. *Communications of the ACM* June 2006; **49**(1): 101–106.
23. Vakali A, Pallis G. Content delivery networks: status and trends. *IEEE Internet Computing* 2003; **7**(6): 68–74.
24. Agyapong P, Sirbu M. Economic incentives in information- centric networking: implications for protocol design and public policy. *IEEE Communications Magazine* 2012; **50**(12): 18–26.
25. Shen Z, Luo J, Zimmermann R, Vasilakos AV. Peer-to-peer media streaming: insights and new developments. *Proceedings of the IEEE* 2011; **99**(12): 2089–2109.
26. Mu M, Ishmael J, Knowles W, Rouncefield M, Race N, Stuart M, Wright G. P2P-based IPTV services: design, deployment, and qoe measurement. *Transactions on Multimedia* 2012; **14**(6): 1515–1527.
27. Saucez D, Grieco LA, Barakat C. AIMD and CCN: past and novel acronyms working together in the Future Internet. In *Proceedings of the 2012 ACM Workshop on Capacity Sharing*, CSWS '12. ACM: New York, NY, USA, 2012; 21–26.
28. Koponen T, Chawla M, Chun BG, Ermolinskiy A, Kim KH, Shenker S, Stoica I. A data-oriented (and beyond) network architecture, In *Proceedings of Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM*, New York, NY, USA, 2007; 181–192.

29. Fotiou N, Nikander P, Trossen D, Polyzos GC. Developing information networking further: from PSIRP to PURSUIT, In *Proceedings of Int. Conf. on Broadband Communications, Networks, and Systems, BROADNETS*, Athenes, Greece, 2010; 1–13.
30. Ahlgren B, et al. Final NetInf architecture, 2012. SAIL FP7 Project, Deliverable D.B.3.
31. Anadiotis AC, et al. Final protocol architecture, 2012. CONVERGENCE Deliverable. D5.3.
32. Bari M, Chowdhury S, Ahmed R, Boutaba R, Mathieu B. A survey of naming and routing in information-centric networks. *IEEE Communications Magazine* 2012; **50**(12): 44–53.
33. Smetters DK, Jacobson V. Securing network content. *PARC Technical Report TR-2009-1*, 2009.
34. Varvello M, Perino D, Esteban J. Caesar: a content router for high speed forwarding, In *Proceedings of ACM Workshop on Information-Centric Networking, ICN*, Helsinki, Finland, 2012; 73–78.
35. Jacobson V, Smetters DK, Briggs NH, Plass MF, Stewart P, Thornton JD, Braynard RL. VoCCN: voice-over content-centric networks, In *ACM ReArch '09*, Rome, Italy, 2009; 1–6.
36. Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E. SIP: session initiation protocol, 2002.
37. Schulzrinne H, Casner S, Frederick R, Jacobson V. *RTP: A Transport Protocol for Real-Time Applications*. RFC Editor: United States, 2003.
38. Handley M, Jacobson V. SDP: session description protocol, 1998.
39. Mumble open source project website, June 2013. [OnLine] Available at: http://mumble.sourceforge.net/.
40. CCNx software, June 2013. Available from: http://www.ccnx.org/.
41. Stockhammer T. Dynamic adaptive streaming over http –: standards and design principles, In *Proceedings of the ACM Conf. on Multimedia Systems, MMSYS*, New York, NY, USA, 2011; 133–144.
42. Sodagar I. The MPEG-DASH standard for multimedia streaming over the Internet. *IEEE Transactions on Multimedia* 2011; **18**(4): 62–67.
43. Awiphan S, Muto T, Wang Y, Su Z, Katto J. Video streaming over content centric networking: experimental studies on PlanetLab, In *Proceedings of IEEE Conf. on Computing, Communications and IT Applications, ComComAp*, Hong Kong, China, 2013; 19–24.
44. Piro G, Ciancaglini V. Enabling real-time TV services in CCN networks. *IEEE COMSOC MMTC E-letter* 2013; **8**(4): 17–20.
45. A simulation platform modelling tv services over the ccnsim, June 2013. Available from: http://telematics.poliba.it/ccn-tv.
46. Carofiglio G, Gallo M, Muscariello L, Papalini M. Multipath congestion control in content-centric networks, In *IEEE NOMEN Workshop, co-located with INFOCOM*, Turin, Italy, 2013.
47. Afanasyev A, Moiseenko I, Zhang L. ndnSIM: NDN simulator for NS-3. *Technical Report NDN-0005*, NDN, 2012.
48. Rossini G, Rossi D. Large scale simulation of ccn networks, In *Algotel 2012*, La grande Motte, France, 2012.
49. A DCE module for the NS-3 simulator enabling the execution of CCNx, June 2013. Available from: http://www-sop.inria.fr/members/Frederic.Urbani/ns3dceccnx/.
50. CCN-lite official page, 2013. Available from: http://www.ccn-lite.net/.
51. Cianci I, Grieco L, Boggia G. Ccn - java opensource kit emulator for wireless ad hoc networks, In *Proceedings of ACM Int. Conf. on Future Internet Technologies, CFI*, Seoul, Korea, 2012; 7–12.

INVITED PAPER

Information-centric networking and multimedia services: present and future challenges

Giuseppe Piro, Luigi Alfredo Grieco, Gennaro Boggia and Periklis Chatzimisios



Because of the high use of multimedia systems among Internet users, significant scientific efforts are being devoted to the investigation of solutions based on information-centric networking. In order to offer a unifying view on such valuable contributions, this paper presents a thorough review of the state of the art, a comprehensive framework to classify the most significant proposals, an overlook at the road ahead towards mature ICN-based multimedia systems and a critical discussion on available tools.