Exploitation of Information Centric Networking Principles in Satellite Networks

A. Detti, A. Caponi, N. Blefari-Melazzi
CNIT - Department of Electronic Engineering, University of Rome “Tor Vergata”
Via del Politecnico 1, Rome (Italy)
{andrea.detti, alberto.caponi, blefari}@uniroma2.it

Abstract— **This paper explores possible advantages of the Information Centric Networking (ICN) paradigm in a geostationary satellite network. We find out that, with respect to plain HTTP services, ICN makes possible to reduce the downstream bandwidth consumed for Internet access by better exploiting the temporal locality of references within requested streams of Web contents. We present an ICN satellite architecture, describe its peculiar mechanisms and assess our solution through simulations.**

Keywords—Information centric network; satellite network; caching; data-centric security; network architecture.

I. Introduction

Information Centric Networking (ICN) is a new paradigm in which the network layer provides users directly with content, instead of providing communication channels between hosts, and is aware of the name (or identifiers) of the contents [1][2][3]. ICN addresses content items by using names that do not include reference to content location.

The data units are *self-contained* and *content-based*, i.e. contain all the information needed to identify the requested or carried content, at least in terms of the content name. For instance, the ICN architectures proposed in [1][2] use Interest messages to request chunks of a content and Data messages to transport chunks of a content. Both messages include the name of the referenced content and chunks.

An ICN routes-by-name [4][7][16] content requests (i.e. Interest messages) towards the “closest” copy of the content. This copy could be stored in the original server, in a cache contained in a network node or even in another user’s device. When the serving device is reached, the ICN delivers the requested content (e.g., Data message) back to the requesting user; intermediate, traversed, nodes may cache the content. Reverse routing of data uses previous-hop information, which is temporary left in the nodes traversed during the content request forwarding process. In doing so, the network path followed by a content request will be used in the reverse direction to transfer the related content.

In ICN, each caching node (not only the original source) may provide the content,; thus, the user needs to trust the content rather than the source. For this reason, ICN adopts *secure* data units, in the sense that security information enabling the verification of the content validity are included in the data unit itself [26]. Traditional networks, instead, secure the channel (connection-based security) or the application (application-based security).

ICN could be exploited in several different network contexts such as wired [12][16][33], wireless and sensors [8][9][10], publish-subscribe [11][13][20], open flow infrastructures [5][14][15], opportunistic networks [15], satellite networks [23].

In this paper we investigate the possible advantages of ICN in a satellite network that provides users with Internet access through a geostationary satellite and a terrestrial gateway station. For instance, this is a typical scenario of a DVB-S2 system [17].

We find out that the use of self-contained, content-based data units, joined with temporal locality of references within the streams of content request [22], can be exploited to reduce the downstream traffic, i.e. from gateway station to terminals.

An ICN data unit describes itself in terms of “what” is requested by or transported in, and such awareness may be used by:

i) a terminal, to easily overhear-and-cache data items transmitted in the downstream path

ii) a gateway station, to serve with a single downstream transmission, concurrent requests of the same data item, as it may occurs in the case of on-demand live video streaming [8].

We observe that overhear-and-cache is difficult in case of traditional HTTP communications. Indeed, an HTTP cache (proxy) needs to be traversed both by upstream and downstream TCP/IP data units and, in a geostationary satellite scenario, overhearing upstream traffic in not practical. Therefore, in a traditional HTTP case, a terminal could cache its local traffic only, and this limitation reduces caching performance.

Moreover, also in case of on-demand live streaming events, traditional HTTP is difficult to exploit, to save downstream bandwidth. It is not easy to serve different HTTP clients of the same live event, by transmitting a single, shared, HTTP stream through the downstream channel, since HTTP is connection-oriented. The two mechanisms just mentioned for exploiting the self-contained content-based data units are not the only possible ones. For instance, it would be possible to implement content-based service differentiation in the gateway, to optimize video streaming quality [18][19].

This paper is organized as follows: in section II we present the satellite ICN architecture. In section III we describe how the satellite ICN exploits temporal locality of content requests to reduce downstream traffic. In section IV we present simulation results and our conclusions.

II. Satellite ICN

In this section we first recall the basic way of operation of an ICN [1][2]. Then we discuss the transport mechanism used to download content [6] and, finally, we present our satellite ICN architecture.

Fig. 1 shows the data model that we consider. A content has a unique network identifier, e.g. "cnn.com/text1"; it is segmented in different chunks and each chunk has an identifier. The chunk identifier includes the content name and the chunk number, e.g. "cnn.com/text1/chunk2". Each chunk is transported by the network within an ICN data unit, called Data message.

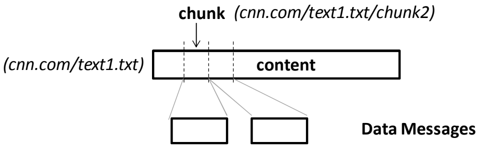


Fig. 1 – Data model

Fig. 2 depicts the exchange of messages during a content download. A client downloads a content by retrieving each chunk. To download a chunk, a client sends out an Interest message that contains the chunk name. The ICN routes the Interest message toward the server, which sends back the data item within a Data message. To download the whole set of chunks, the client adopts a TCP-like receiver-driven approach [6], where TCP ACKs are replaced by Interest messages; TCP segments are replaced by Data messages, and traffic control operation is carried out at the receiver side, by using a TCP-like congestion window (cwnd) control, applied on the number of in-flight Interest messages.

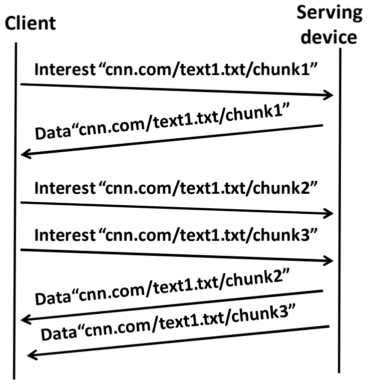
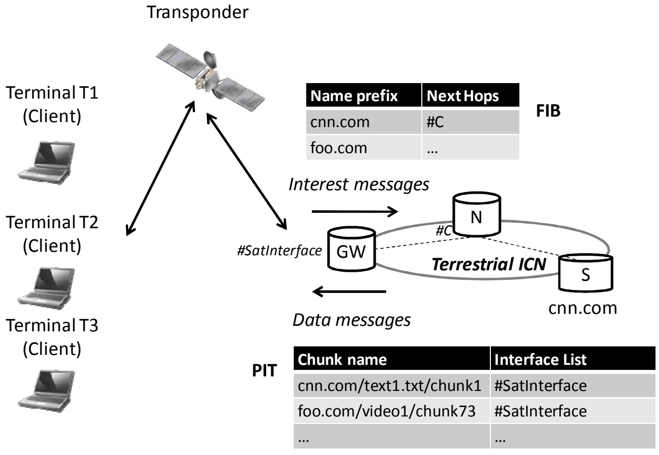

Client
Serving device
Interest "cnn.com/text1.txt/chunk1"
Data "cnn.com/text1.txt/chunk1"
Interest "cnn.com/text1.txt/chunk2"
Interest "cnn.com/text1.txt/chunk3"
Data "cnn.com/text1.txt/chunk2"
Data "cnn.com/text1.txt/chunk3"

Fig. 2 – Download of a content

For instance, at the start of the communication, the client has a cwnd equal to 1 and transmits Interest for chunk 1. When chunk 1 is received, cwnd is increased by 1, as specified by the slow-start TCP algorithm. Consequently, the client transmits two Interest messages, for chunk 2 and 3. When these Data messages will be received, cwnd will be increased again, next Interest messages will be generated, and so forth.

Fig. 3 shows the satellite ICN network. We have satellite terminals, which are the ICN client nodes that request contents. Clients send Interest messages to the gateway station (GW). The gateway station is an ICN node connected to a terrestrial ICN, which contain Servers (e.g., S) and generic network nodes (e.g., N).



Name prefix	Next Hops
cnn.com	#C
foo.com	…

Chunk name	Interface List
cnn.com/text1.txt/chunk1	#SatInterface
foo.com/video1/chunk73	#SatInterface
…	…

Fig. 3 – Satellite ICN

To route an Interest message, a generic ICN node uses a name-based Forwarding Information Base (FIB), whose entries are in the form <name-prefix, next hops>. A longest matching algorithm, based on characters, selects the best entry of the FIB. For instance, in case of Fig. 3, when the GW node receives from the satellite interface an Interest for "cnn.com/text1.txt/chunk1", then the longest match selects the FIB entry "cnn.com". Then the GW node forwards the Interest message towards the next ICN interface of the path, i.e. #C. We observe that, depending on the under-ICN technology, #C could be a UDP/IP socket, an Ethernet address, etc.

During the forwarding of an Interest message, an ICN node temporarily stores the couple <chunk name, receiving interface list> in a Pending Information Table (PIT). The PIT contains information about the set of Interest messages received by a node and not yet served, i.e. messages for which the node has not yet sent back the related Data message.

PIT entries are grouped by chunk name and the interface list field contains the set of ingress interfaces that received the related Interest messages.

If the chunk name of a received Interest is already included in a PIT entry, then the incoming interface is added (if absent) to the interface list of the PIT entry. Then the Interest is not forwarded, to avoid the duplication of Interest/Data messages in case of concurrent request of a same chunk.

When a node receives a Data message, it lookups the chunk name in the PIT, it forwards Data message towards all the interfaces contained in the interface list and deletes the PIT entry.

We point out that, by using the PIT mechanism, the gateway station serves concurrent requests of the same chunk with a single downstream transmission of a Data message.

In the scenario reported in Fig. 3, when the gateway station receives an Interest message for "cnn.com/text1.txt/chunk1" from the satellite interface, the node stores the entry < cnn.com/text1.txt/chunk1, {#SatInterface}> in the PIT. In case a second Interest message for "cnn.com/text1.txt/chunk1", coming from another terminal, is concurrently received by the gateway, the PIT entry is already set and so the second Interest message is discarded. When the gateway station receives the related Data message for "cnn.com/text1.txt/chunk1", then the gateway station lookups the entry in the PIT, forwards the message on the #SatInterface and deletes the PIT message. After propagation time, both terminals will receive the required Data message.

An ICN node may cache received data items in a local memory. In this case, when the node receives an Interest message, first it checks the presence of the related Data message in its cache. If a cache hit occurs, the node sends back the cached data item. In case of cache miss, the node executes the forwarding and PIT operations previously described.

We observe that the caching of fake contents could be very risky as it may lead to a critical Denial of Service, by preventing the download of an original content [24]. For this reason, in our architecture, a node verifies the validity of the cached data, before caching a data item, by using control information contained in the Data message. This information includes the public key of the principal of the resource and a digital signature. To limit the overhead due to the transport of security control information, we envisage the use of signature techniques based on elliptic curves, such as ECDSA [29], or Identity Base Signature [28][29]. As a drawback, the use of elliptic curves increases the processing load and this issue should be taken into account for hardware dimensioning.

Up to now we described well-know ICN operations (e.g., [1][2]). In what follows we introduce two specific satellite mechanisms that improve the exploitation of temporal locality of references within Web request traces.

Overhearing – Usually, an ICN node caches Data messages for which the node is a next-hop or it is the client. For satellite terminals we expanded the cacheable data set, by enabling a Terminal node to overhear and cache any Data message sent on the downstream path. For instance, if the gateway station sends a Data message "cnn.com/text1.txt/chunk1" towards terminal T1, any other terminal caches the data item.

Delayed operations on the gateway station – The Terminal to Gateway path has a propagation delay of about 250ms. In case of content requests close in time between each other, such delay could make caching operations not effective. Fig. 4 reports an example describing this issue.

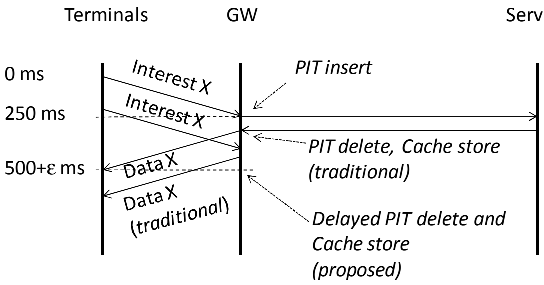


Fig. 4 – Delayed operations

At time 0ms, terminal T1 sends an Interest message for chunk X. The Interest message reaches the gateway station at time 250ms. The gateway station inserts an entry in the PIT, routes-by-name the Interest X toward the Server and, after ε ms, receives back the related Data message. Considering that the Data messages are relatively small (e.g. 4 kbytes) and terrestrial network delays are relatively short, we could assume the delay ε practically negligible with respect to 250ms. Upon the reception of the Data message, the gateway station deletes the PIT entry, and caches and sends the message through the downstream channel. Data message will be received by all Terminals (overhearing) at time 500ms + ε.

Let us assume that at a time instant comprised between ε and 250 ms, another terminal T2 wishes to download the same chunk X. At this time the Data message is not yet overheard by terminal T2, hence chunk X is not present in the local cache of T2. Consequently, T2 sends out an Interest for chunk X (second Interest of the figure). This second Interest reaches the gateway station, which has the message in its cache and sends back the Data message (the second Data of the figure) through the downstream path. Therefore, in this case we have two downstream transmissions of the same data item, and overhearing has been useless.

To avoid the duplication of downstream transmissions, we must prevent the gateway station to retransmit data items that are "in-flight". Accordingly, we propose to delay by the propagation period (250ms) both the deletion of a PIT entry and the caching of a Data message. In doing so, when the second Interest message is received, the gateway station has a cache miss and still has the PIT entry for X. Consequently, the Interest message is discarded and nothing is transmitted, neither towards the server nor on the satellite downstream path. Anyway, the terminal T2 will receive the requested Data message X by overhearing messages previously transmitted for terminal T1.

III. Exploiting the Temporal Locality

Literature studies on real HTTP traces show that references in Web request streams present spatial and temporal locality [22].

There are two kinds of spatial locality, which we can name client-side and server-side. Client-side spatial locality is due to the geographic correlations of the Web surfing behavior: geographically close users are interested in similar contents. In the considered ICN satellite network, we assume that the network is managed by a service provider, which mainly serves geographically close users, e.g. users of the same country. Therefore, spatial locality is already partially exploited and in this paper we do not include techniques to further exploit it. Anyhow, we briefly observe that different approaches could be deployed, to exploit client-side spatial locality. For instance, the authors of [23] propose that a network operator creates profiles of users and that terminals cache only Data messages requested by users with a close profile.

Server-side spatial locality occurs when there is a temporal correlation among requests of Web objects contained in the same Server. This correlation is related to the structure of Web pages, which are usually formed by different objects, stored in the same server. Server-side spatial locality could be used to pre-fetch contents on terminal caches [22]. Nevertheless, by pre-fetching we could risk to waste satellite bandwidth that, conversely, is what we want to save.

Temporal locality shows up when requests of the same content are close in time. Such a temporal correlation is usually modeled through an Least Recently Used (LRU) stack distance model [22][25].

Our satellite ICN network exploits temporal locality as follows. Depending on the distribution of reuse time distance, we identify two coarse regions of temporal locality, namely *standard* and *strong*.

In case of standard temporal locality, the reuse distance time between two consecutive requests of the same data item is: i) greater than the download time, and ii) shorter than the time spent by an item in a cache. In this case, caching is effective in reducing consumed bandwidth, since data are likely present in the cache when next requests are issued. Furthermore, after that terminal T1 has downloaded a data item, the probability that a request of the item will be submitted again by T1 is lower than the probability that the request is submitted by any other terminal. Therefore, it is convenient to cache the data item at any terminal by overhearing.

In presence of strong temporal locality, several users request the same data item almost at the same time. For instance, this may occur during live streaming transmissions [8], with strict delay requirement. In this case, caching is not effective since, when the same Interests are issued by different terminals, the Data message is not yet completely transferred on the satellite path and, therefore, not yet cached at terminal stations. Opportunely, this "storm" of Interest messages is blocked at the gateway station node, which stores in the PIT a single entry for the data item and forwards towards the server only the first received Interest message. After that, when the Data message will come back, only one copy will be transferred on the satellite interface.

We conclude that:

i) in case of standard temporal locality, satellite bandwidth is saved by overhear-and-cache;

ii) in case of strong temporal locality, satellite bandwidth is saved at the gateway station by the PIT mechanism.

IV. Performance assesment

In this section we carry out a performance evaluation aimed at showing the effectiveness of our satellite ICN network in saving downstream bandwidth.

We observe that, in case of traffic with strong temporal locality, the bandwidth saving is obtained "*by construction*" using PIT mechanisms [1]. Therefore, we limit our investigation to the effectiveness of overhear-and-cache in cases of traffic with standard temporal locality.

Our work answers to the following question: "in case of (standard) temporal locality, is overhearing better than no-overhearing ?"

In this paper we do not evaluate "how much" is the gain provided by overhearing, since it obviously depends on (arguable) traffic parameters used in the simulations. As a preliminary work, our focus here is limited to investigate whether "there is" a gain. The availability of real traces will allow us to evaluate the amount of gain in realistic scenarios. This we will do in future works.

We used the ccnSim framework [31] to simulate ICN functionalities, i.e., forwarding, caching, PIT operations, etc. We used the tool proWGen [27] to generate a traffic trace. This tool models temporal locality by using an LRU stack, the content popularity by using the traditional Zipf distribution [21], and content request inter-arrival time by using an exponential distribution. Tab. 1 reports the main parameters of the proWGen configuration.

Item	Value
Number of contents	116.301
Number of chunks	370.326
Avg. content size (kB)	13,34
Zipf slope	0,75
One-timers	0,7
LRU depth	100

Tab. 1 - Parameters of the trace with standard temporal locality

We also generated a trace *without* temporal locality, so as to evaluate the effectiveness of overhear-and-cache both in presence and in absence of temporal locality.

We generated the trace without temporal locality as follows. A trace is a matrix of two columns <request_time, content_id>. The first column is the sequence of request times, while the second column is the identifier of the requested contents. To generate the trace without time locality, we start from the trace with temporal locality and randomly scramble the elements of the second column. In doing so, we maintain the same popularity distribution in the two traces and remove temporal correlation.

The network simulation scenario consists of 50 terminals and of a gateway station, directly connected to a terrestrial server. The terminal to gateway one-way delay is 250 ms and the shared downstream transmission bandwidth is 50 Mbit/sec. The delay and bandwidth among gateway station and server is zero and infinite, respectively. Content requests contained in a trace are uniformly distributed among terminals. Chunk size is 4192 bytes. Cache replacement policy is the traditional Least Recently Used (LRU). At the simulation start, caches are prefilled with popular contents, so reducing the time needed to reach steady state conditions.

Fig. 5 reports the ratio between the downstream traffic rate sent in case of overhearing and the same rate in case of no-overhearing. We remind that, without overhearing, a terminal stores in the cache only its own Data messages.

In these simulations we varied the content request frequency and evaluated the performance obtained in presence and in absence of temporal locality. Terminals cache size has been fixed to 4000 chunks, i.e. about 1% of the whole set of chunks.

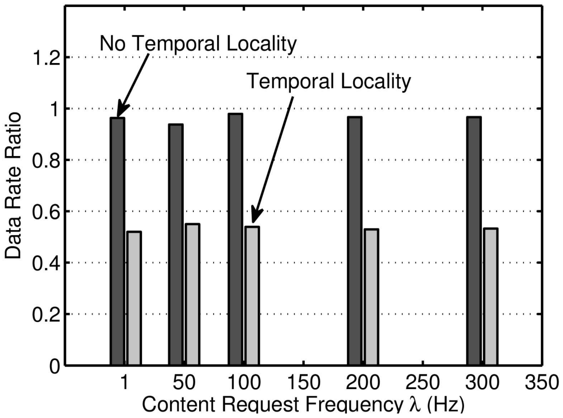


Fig. 5 – Ratio of the downstream data rates versus the cumulative rate of content request per second (λ)

Fig. 5 shows that in presence of temporal locality the ratio is not greater than 0.54; hence, overhearing yields a reduction of downstream traffic up to 46%. Conversely, overhearing does not improve performance in absence of temporal locality. Such performance is stable by varying the request frequency. This insensitiveness is due to the “delayed operation” at the gateway station described in Section II. Indeed, without this mechanism, preliminary simulation results showed a valuable performance decrease at the increase of content request frequency, because of the increasing occurrence of the event described in Fig. 4.

Fig. 6 shows the same ratio of Fig. 5 but this time vs. the cache size. Here we used a content request frequency of 50Hz. Also in this case we observe a stable performance. Anyhow, albeit not reported, in the extreme cases of absence of cache and of a cache with a size equal to the overall number of chunks, the data rate ratio obviously tends to 1.

A. Feasibility of the validity check

As discussed in section II, a terminal has to verify the integrity and authenticity of Data message, before caching it. To verify the feasibility of this operation we measured the time needed to verify Identity Based Signatures (IBS) [29] and ECDSA signatures [30].

Well-known IBS schemes are based on *pairing cryptography* and provide low overhead for the signature, but have a slow verification process. In [28] the authors proposes an IBS scheme using Schnorr signatures concatenation, instead of pairing cryptography, making this scheme very competitive, compared with more traditional signatures (i.e. RSA, DSA, ECDSA, etc.).

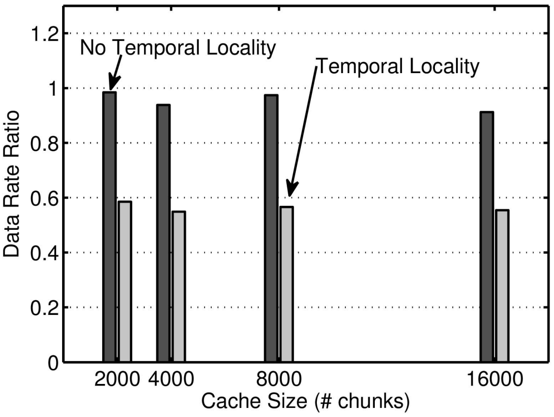


Fig. 6 – Ratio of the downstream data rates versus the cache size

Such scheme provides verification delay close to traditional signatures and low signature overhead (certificate transmission is never needed). We compared the IBS signature scheme to a traditional ECDSA scheme, both implemented using OpenSSL API and using an Intel I7 processor @1.8Ghz.

Tab. 2 reports the time required to sign and verify a chunk of 4kB by using 160-bit elliptic curves. These values enable a verification rate of about 120Mbps using ECDSA (4kB/0.28ms), and about 90Mbps using IBS scheme. Such bitrates are close (or even greater than) to the rate supported by current satellite transponders, e.g. based on DVB-S2 technology. Therefore, overhear-and-cache seems practical.

Scheme	Verification (ms)	Rate (Mpbs)
ECDSA	0.28	120
IBS [28]	0.36	91

Tab. 2 – Signature verification time of the ECDSA and IBS schemes.

V. CONCLUSIONS

Applying Information Centric Networking paradigm to a geostationary satellite network can lead to bandwidth savings, greater than those obtained with traditional HTTP means. ICN content-based data-units enable the implementation of techniques such as overhear-and-cache and Pending Information Table. These techniques save bandwidth by exploiting the temporal locality of references within request streams of Web objects, and are deemed to be effective also in case of live streaming events. The implementation of such functionalities seems to be more difficult by using traditional HTTP, due to its connection-oriented nature.

ACKNOWLEDGMENT

This work is partially funded by the EU (FP7 CONVERGENCE project [32]).

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. Briggs, R. Braynard, ”Networking named content”, ACM CoNEXT 2009

[2] A. Detti, N. Blefari Melazzi, S. Salsano, M. Pomposini, “CONET: A Content Centric Inter-Networking Architecture”, ACM SIGCOMM Workshop on Information-Centric Networking (ICN 2011), August 19, 2011, Toronto, Canada

[3] D. Trossen, M. Sarela, and K. Sollins: "Arguments for an information-centric internetworking architecture" SIGCOMM Computer Communication Review, vol. 40, pp. 26-33, 2010

[4] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, “A data-oriented (and beyond) network architecture”, ACM SIGCOMM 2007

[5] N. Blefari Melazzi, A. Detti, G. Morabito, S. Salsano, L. Veltri: “Supporting Information-Centric Functionality in Software Defined Networks”, IEEE ICC 2012, Software Defined Networks Workshop, June 10-15, 2012 Ottawa, Canada

[6] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, N. Blefari-Melazzi, “Transport-layer issues in Information Centric Networks”, ACM SIGCOMM Workshop on Information-Centric Networking (ICN 2012), August 17, 2012, Helsinki, Finland

[7] N. Blefari Melazzi, A. Detti, M. Pomposini, S. Salsano: “Route discovery and caching: a way to improve the scalability of Information-Centric Networking”, IEEE Global Communications Conference 2012 (Globecom 2012), December, 3-7 2012, Anaheim, California

[8] A. Detti, M. Pomposini, N. Blefari Melazzi, S. Salsano, A. Bragagnini, “Offloading cellular networks with Information-Centric Networking: the case of video streaming", 13th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, IEEE WoWMoM 2012, San Francisco, California, USA June 25-28, 2012.

[9] F Cuomo, E Cipollone, A Abbagnale: “Performance analysis of IEEE 802.15. 4 wireless sensor networks: An insight into the topology formation process”, Computer Networks 53 (18), 3057-3075.

[10] U Monaco, F Cuomo, T Melodia, F Ricciato, M Borghini: “Understanding optimal data gathering in the energy and latency domains of a wireless sensor network”, Computer Networks 50 (18), 3564-3584.

[11] L. Chiariglione, A. Difino, N. Blefari Melazzi, S. Salsano, A. Detti, G. Tropea, A. C. G. Anadiotis, A. S. Mousas, I. S. Venieris, C. Z. Patrikakis: “Publish/Subscribe over Information Centric Networks: a Standardized Approach in CONVERGENCE”, Future Network & Mobile Summit 2012, 4 - 6 July 2012, Berlin, Germany

[12] N. Blefari Melazzi, A. Detti, M. Pomposini: “Scalability Measurements in an Information-Centric Network”, in “Measurement-based experimental research: methodology, experiments and tools”, Springer Lecture Notes in Computer Science (LNCS), vol. 7586, 2012, Editors: Lluís Fàbrega, Pere Vilà, Davide Careglio, Dimitri Papadimitriou

[13] P. K. Gkonis, C. Z. Patrikakis, A. G. Anadiotis, D. I. Kaklamani, M. T. Andrade, A. Detti, G. Tropea, N. Blefari Melazzi: “A Content-Centric, Publish-Subscribe Architecture delivering Mobile Context-Aware Health Services”, Future Network and Mobile Summit 2011, Warsaw, Poland, 15-17 June 2011

[14] N. Blefari Melazzi, A. Detti, G. Mazza, G. Morabito, S. Salsano, L. Veltri: “An OpenFlow-based Testbed for Information Centric Networking”, Future Network & Mobile Summit 2012, 4 - 6 July 2012, Berlin, Germany

[15] G. Bianchi, P.Loreti and A.Trkulja,“Let me grab your app:A preliminary proof-of-concept design of opportunistic content augmentation,” in Third IEEE Workshop on User-Centric Networking, 2012

[16] A. Detti, M. Pomposini, N. Blefari Melazzi, S. Salsano: “Supporting the Web with an Information Centric Network that Routes by Name, Computer Networks: The International Journal of Computer and Telecommunications Networking, Elsevier North-Holland, Inc. New York, NY, USA, http://dx.doi.org/10.1016/j.comnet.2012.08.006

[17] Digital Video Broadcasting (DVB), “Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)”, ETSI EN 302 307, V1.2.1, April 2009

[18] A. Detti, G. Bianchi, P. Loreti, C. Pisa, F. S. Proto, S. Thakolsri, V. Kellerer, J. Widmer, ”SVEF: an Open-Source Experimental Evaluation Framework for H.264 Scalable Video Streaming”, IEEE MediaWiN 2009

[19] G. Bianchi, A. Detti, P. Loreti, C. Pisa, S. Thakolsri, V. Kellerer, J. Widmer, “Cross-layer H.264 Scalable Video Downstream Delivery Over WLANs”, IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2010 (WoWMoM 2010), Montreal, Canada, 14-17 June 2010

[20] A. Carzaniga, M. Papalini and A. Wolf, ”Content-Based Publish/Subscribe Networking and Information-Centric Networking”, ACM SIGCOMM 2011, ICN workshop

[21] L. Breslau, P. Cao, L. Fan, G. Phillips. S. Shenker, “Web Caching and zipf-like Distribution: Evidence and Implications”, IEEE INFOCOM 1999

[22] V. Almeidats, A. Bestavros, M. Crovellat, A. Oliveiraj, “Characterizing Reference Locality in the WWW”, Parallel and Distributed Information Systems, 1996

[23] L. Galluccio, G. Morabito, S. Palazzo, “Caching in information-centric satellite networks”, IEEE International Conference on Communications, ICC 2012

[24] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, and J. Wilcox, "Information-Centric Networking: Seeing the Forest for the Trees", in Proc. of the 10th ACM Workshop on Hot Topics in Networks (HotNets-X), Cambridge, Massachusetts

[25] R. Mattson, J. Gecsei, D. Slutz, I. Traiger. Evaluation techniques for storage hierarchies. IBM Systems Journal, 9(2):78–117, 1970

[26] D. Smetters, V. Jacobson, “Securing Network Content”, PARC Tech Report, October 2009

[27] M. Busari, C. Williamson, “ProWGen: a synthetic workload generation tool for simulation evaluation of web proxy caches”, *Comput. Netw*. 38, 6 (April 2002), 779-794.

[28] D. Galindo, F. D. Garcia, “A Schnorr-Like Lightweight Identity-Based Signature Scheme”, in *Proceedings of the 2nd International Conference on Cryptology in Africa: Progress in Cryptology* (AFRICACRYPT '09), Bart Preneel (Ed.). Springer-Verlag, Berlin, Heidelberg, 135-148.

[29] Adi Shamir, “Identity-Based Cryptosystems and Signature Schemes”, Advances in Cryptology: Proceedings of CRYPTO 84, Lecture Notes in Computer Science, 7:47--53, 1984

[30] D. Johnson and A. Menezes, “The elliptic curve digital signature algorithm (ECDSA)”, Technical Report CORR 99-34, Department of Combinatorics & Optimization, University of Waterloo, Canada, February 24 2000

[31] ccnSim: scalable chunk-level simulator of Content Centric Networks, http://perso.telecom-paristech.fr/~drossi/index.php?n=Software.ccnSim

[32] CONVERGENCE website: www.ict-convergence.eu

[33] D. Trossen, G. Parisis, Designing and Realizing An Information-Centric Internet, IEEE Communications Magazine, Special Issue on Information-centric Networks, July 2012