I-CAN: Information-Centric Access Networking*

Architecture and Experiments

Vasilios A. Siris, Nikos Fotiou, Dimitrios Dimopoulos, George C. Polyzos
Mobile Multimedia Laboratory
Department of Informatics, School of Information Sciences and Technology
Athens University of Economics and Business, Patision 76, 104 34, Athens, Greece
{vsiris, fotiou, dimdimopoulos, polyzos}@aueb.gr

***Abstract*—We present the Information-Centric Access Network (I-CAN) architecture, which is based on the publish-subscribe Information-Centric Networking (ICN) paradigm, identifying how it accounts for specific characteristics of mobile and wireless access networks. We also present initial results from the testbed implementation of two application scenarios that exploit key features of the I-CAN architecture: secure publication proxy and multi-source mobile video streaming.**

Keywords—Information-Centric Networks; security; mobile video streaming; multi-source data transfer

I. INTRODUCTION

Mobile traffic in 2014 grew by 69%, becoming nearly 30-times the global Internet traffic in 2000, and is expected to grow 10-fold from 2014 until 2019[1]. A promising solution to address the strain from the exponential growth of mobile traffic is to move a portion of it to Wi-Fi networks, exploiting the existence of multiple wireless interfaces in smartphones (i.e., both 3G/4G and Wi-Fi) and the significantly lower cost of Wi-Fi technology. At the same time, the Internet's current dominant usage model involves end-users exchanging information or accessing services, independently of the device that provides them. Moreover, not only the consumption but also the production of content is becoming user-centric, requiring a network infrastructure that facilitates the efficient delivery of user-generated content and considering the connectivity and energy constraints of mobile sources.

The goal of the I-CAN project is to develop and evaluate architectures and procedures for future access networks based on Information-Centric Networking (ICN) in order to radically advance the integration of cellular (licensed spectrum) and wireless (license-exempt) access technologies. ICN decouples the data from the actual devices storing it through the location-independent naming of content. This decoupling presents a fundamental departure from the Internet's host-centric communication model towards an architecture that matches the Internet's current dominant usage identified above.

Content-awareness along with the decoupling of content creation, advertisement, and transfer, offers opportunities for enhancing both receiver and content mobility. For user-generated content, mobile sources can announce the availability of content, deferring its actual transfer until it is requested; such an approach can have benefits in terms of cost, energy efficiency, and improved privacy and access control.

The possibility of ubiquitous in-network caching, which is inherent to ICN due to its explicit naming of content rather than communication endpoints, opens up many opportunities for exploiting content-awareness in order to place information closer to the user. Moreover, by naming the content itself, ICN allows the receiver to obtain different parts of the content from different sources and through different paths. Optimizations in some of the above directions through overlay solutions to the current Internet are possible, but such solutions can be costly and are necessarily application-dependent. A key direction of ICN is to create a networking substrate that provides application-independent communication primitives that satisfy emerging information transfer and communication requirements, thus facilitating a more efficient and timely development of distributed data sharing applications, which is crucial for future innovations in the mobile space.

The contribution of this paper is to present the I-CAN architecture, which is based on the Publish-Subscribe Internet (PSI) architecture developed in the PURSUIT project [1], identifying how it accounts for specific characteristics of mobile and wireless access networks, and discuss initial results from the testbed implementation of two application scenarios that exploit key features of the I-CAN architecture: secure publication proxy and multi-source mobile video streaming.

II. THE PUBLISH-SUBSCRIBE INTERNET ARCHITECTURE

The PSI [1] architecture is based on the publish/subscribe paradigm where users interested in receiving some content *subscribe* for it through their network device, referred to as the *subscriber*, and content *owners* store their content on a network device which *advertises* it and if requested *publishes* it (hence these devices are referred to as the *publishers*).

Every content item is identified by a flat identifier known as the *Rendezvous Identifier* (RId). Moreover, every content item belongs to at least one *scope*. The purpose of a scope is to give a hint about content location and to group content items with the sane dissemination level. Scopes are hierarchically

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019, Feb. 5, 2015.

* This work was co-financed by the EU (European Social Fund–ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF)–Research Funding Program: Aristeia II/I-CAN.

organized and identified by a *Scope Identifier* (SId). Scopes are managed by specialized *Rendezvous Nodes* (RNs), which form an overlay *Rendezvous Network*. The rendezvous network provides a lookup service, which routes a subscription to a RN that "knows" (at least) one publisher for the requested item.

A typical transaction in PSI involves the following steps (see Fig. 1). An owner of a content item assigns it a RId and stores a copy of it in at least one publisher that advertises its availability in one or more scopes (message 1). With this advertisement, the RId is stored in the RNs that manage these scopes. Subscribers send subscriptions for specific (SId,RId) pairs (message 2), which are routed by the rendezvous network to an appropriate RN. Upon receiving a subscription message and provided that at least one publisher exists, the RN instructs a *Topology Manager* to create a forwarding path from a publisher to the subscriber, which is included in the notification message to the publisher (message 3). Finally, the content item is transferred from the publisher to the subscriber (message 4).

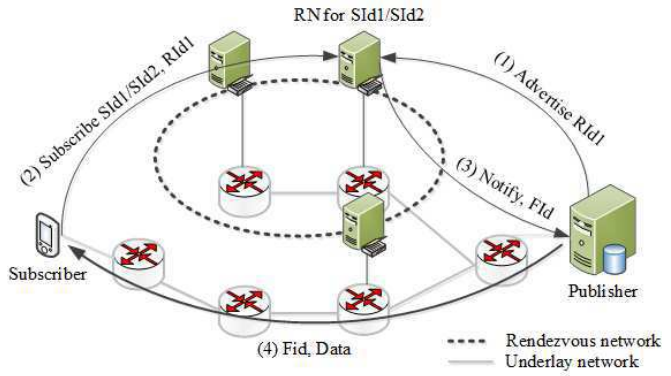

Fig. 1. The Publish-Subscribe Internet architecture

III. THE I-CAN ARCHITECTURE

The I-CAN architecture adapts PSI to an access network. In this section we discuss the main features of the I-CAN architecture, identifying how it accounts for specific characteristics of mobile and wireless access networks.

A. Content types and content naming

I-CAN supports two types of content: channel and document. On the other hand, the PSI architecture supports only channel types of content. Channel content items are tolerant of packet losses, hence are delivered using unreliable transport protocols over unidirectional paths. Live media streaming is an example of a channel content item. Document content items, on the other hand, are delivered using reliable transport protocols over bidirectional paths. A publisher must include the content type in the advertisement messages, since this can be required by the network to decide whether to create unidirectional or bidirectional forwarding paths. Content items with global significance require a unique association between the (SId, RId) pair and the content item. On the other hand, content items with local significance (e.g., local weather notifications) can be identified using a local scope identifier and are not globally unique. Finally, the granularity of a (SId, Rid) pair is application specific, and can range from a file chunk to a whole file. Similarly, the temporal dependency (or independency) of the SId, RId mapping is application specific.

B. Publication proxies and mobile publishers

The I-CAN architecture considers "proxy" publishers that store and advertise content on behalf of actual owners. A proxy publisher is usually a server appointed by the owner, but can also be any device that "happens" to have a piece of content ("opportunistic" proxy) e.g., a (mobile) device that just received the content. The advantage of publication proxies is that the content remains available even when an owner's device is offline and content transfer is not affected by the owner device's mobility (publisher mobility). Alternatively, owners can store content on their own (mobile) device. In this case owners can initially advertise the availability of their content, deferring its actual transfer until more cost and energy efficient connectivity options are available. An additional benefit of this approach is that retaining content on the owner's device allows for improved privacy and access control.

C. Subscription proxies and proactive caching

The I-CAN architecture introduces "proxy" subscribers, which can send subscriptions on behalf of actual receiver devices. This can be beneficial in cases where receivers are mobile and experience periods of disconnection. Moreover, by proactively fetching content, proxies can make content immediately available in local access networks where the requesting device will connect to, such as small cells and hotspots, overcoming the limited capacity of the backhaul links which is typical for such access networks. Finally, proxy subscribers can significantly reduce the overhead of signaling messages, e.g., in the case of mobile applications requesting frequent updates from application servers, which can be significant in cellular networks and lead to signaling storms.

D. Subscriptions to future content & persistent subscriptions

A subscription may concern both content that exists prior to the subscription, e.g., video files, and content that is produced (or changes) at some time after the subscription, e.g., news or twitter updates and alarm notifications. Also, subscriptions may concern content for which, at the time of the subscription, there is no available publisher. For example, a user may subscribe to a piece of content originally hosted in a mobile device which is currently offline; when the device becomes available it will be notified about the pending subscriptions. Moreover, subscriptions can have a persistent nature, i.e., a single subscription can declare a node's interest in receiving all future instantiations of the requested content, e.g., all future notifications corresponding to a particular identifier; this allows push-based content dissemination, in addition to ICN's native pull-based model. A significant benefit of persistent subscriptions is the reduction of the signaling overhead in cellular networks and contention in shared access wireless networks. The persistence of a subscription can be related to the RId, SId semantics and is application specific.

E. Centralized, decentralized, and fast rendezvous

I-CAN considers two approaches for implementing rendezvous: centralized and decentralized. Centralized

rendezvous is implemented using a few nodes (or even a single node within a local domain). These nodes are considered to be well-known and are used by subscribers for content subscriptions and publishers for content advertisements. Centralized rendezvous is the only option supported in the current PSI implementation. With decentralized rendezvous, every network node, including mobile devices, can perform subscription resolution. In this case the content advertisements and/or subscriptions can be broadcasted inside the local network, exploiting the inherent broadcasting support in wireless networks. Also, such an approach can take advantage of emerging device-to-device communication, such as Wi-Fi direct, which do not require infrastructure support. Centralized and decentralized rendezvous are not mutually exclusive; both can be used simultaneously to support multi-source content transfer, as discussed in the next subsection and Section IV.B.

After resolution of the initial subscription, through which receivers obtain the identity of the publishers for the content they requested, the receivers can send subsequent subscriptions directly to publishers. Advantages of such a fast rendezvous include reducing the load on the rendezvous network (in the case of centralized rendezvous), reducing the rendezvous broadcasting overhead and the contention in shared access wireless networks (in the case of decentralized rendezvous), and reducing the latency for obtaining the requested content.

F. Multi-source and multi-interface content transfer

A key feature of ICN architectures is that they allow content transfer from multiple publishers. In I-CAN this is supported by the rendezvous network, which may either select a single publisher or send a list of all (or some) of the available publishers to the subscriber, which then selects the particular publishers to obtain the requested content. Alternatively, the rendezvous network may directly notify the publisher (or a subset of the available publishers) to send the requested content to the subscriber. The above flexibility enables two types of multi-source transfers. First, a subscriber can obtain different parts of a content item from different sources. Second, a subscriber can obtain content simultaneously from multiple sources; for example, a monitoring node can request a particular measurement from a group of sensor nodes. Hence, I-CAN can support one-to-many/any, many/any-to-one, and many/any-to-many/any connectivity.

The I-CAN architecture allows subscribers to efficiently use multiple heterogeneous interfaces and connectivity options, which include 3/4G, infrastructure Wi-Fi, and device-to-device communication such as Wi-Fi direct; the selection and load balancing of content transfer across multiple interfaces should consider their current state (e.g., delay, loss, and throughput), in conjunction with the user/application requirements. Benefits of multi-source and multi-interface transfer include improved resilience against publisher and network failures, efficient utilization of heterogeneous mobile and wireless resources, and improved QoE (Quality of Experience).

G. Security

The I-CAN architecture adopts a content-oriented security model, i.e., it employs security mechanisms for securing content, rather than securing communication channels. A content name in I-CAN may include direct "security bindings" that ensure content integrity and authenticity. Such a direct security binding could be the usage of RIds of the form "owner public key | content hash". Nevertheless, since direct bindings result in content names that are not human readable, indirect bindings are also considered: pre-trusted rendezvous points can be used to map human readable content names to security primitives. I-CAN uses Identity Based Encryption (IBE) [2]. An IBE scheme is a public key scheme where an arbitrary string (including a human readable name) can be used as the public key. A constrain of IBE is that all entities should know some publicly available "system parameters." In an access network system parameters can be easily disseminated, e.g., using a DHCP-like mechanism, or simply by periodically broadcasting them. As discussed in subsections III.B and III.C, proxies are an essential component of the architecture. Nevertheless, proxies introduce security and privacy risks. Traditional end-to-end encryption poses hurdles to proxy-based communication. Consider for example the case of a publisher proxy: if end-to-end encryption is used the content owner has either to share the same encryption key will all subscribers or to generate as many encrypted versions of the content as the number of subscribers. In order to remedy this problem, I-CAN adopts proxy re-encryption (PRE) [3]. PRE allows third-parties (proxies), to re-encrypt a ciphertext, encrypted with the public key of a user A (usually the publisher), in a way that another user B (usually the subscriber) can decrypt it with his own secret private key. The re-encryption process leaks no information to the proxy.

IV. EVALUATION

A. Secure publisher proxy

In this section we present the design and implementation of a secure publisher proxy that combines Identity Based Encryption (IBE) and proxy re-encryption (PRE). This solution is based on the Identity-based PRE scheme proposed in [3] and is composed of the following functions:

- *Setup():* This function is executed by a Private Key Generator (PKG) and outputs public System Parameters (SP) and a private Master Secret Key (MSK).
- *KeyGen():* This function is executed by a PKG, takes as input the MSK and an identity ID, and outputs the secret key SK_{ID} that corresponds to the identity ID.
- *Encrypt():* This function takes as input SP, an identity ID and a message MSG and outputs the encryption C_{ID} of MSG using ID as public key.
- *RKGen():* This function takes as input SP, a secret key SK_{ID1} and an identity ID2 and generates a (public) re-encryption key $RK_{ID1->ID2}$.
- *Reencrypt():* This function takes as input SP, a (public) re-encryption key $RK_{ID1->ID2}$, and a ciphertext C_{ID1} and outputs a new ciphertext CID2.
- *Decrypt():* This function takes as input SP, a secret key SK_{ID} and a ciphertext C_{ID} and outputs the decryption of the ciphertext.

A secure publisher proxy is implemented as follows (Fig. 2). A content owner encrypts content items using a symmetric encryption key (different for each item). Each symmetric key is then encrypted using IBE and the identity of the owner. The encrypted content items and the encrypted symmetric keys are stored in a proxy. To access the encrypted content, a subscriber needs to decrypt the symmetric encryption key. This can be achieved by having the proxy re-encrypt the symmetric key and derive $C_{Subscriber}$ from C_{Owner}. The re-encryption key for this process can only be generated by the content owner. In our solution we consider two approaches: (i) the owner generates all possible re-encryption keys, stores them in the proxy and then goes off-line, or (ii) the owner is online and generates the appropriate re-encryption key for every request. An interesting property of this solution is that an owner can generate a re-encryption key for an identity that does not yet exist, but will be created in the future.

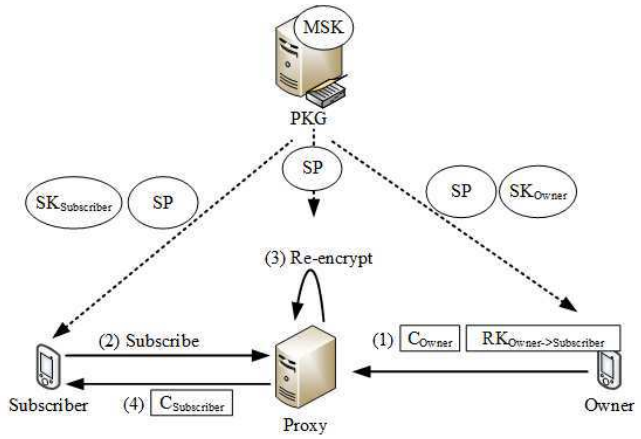

Fig. 2. Secure publisher proxy scenario

Our prototype is implemented using the Charm Crypto library [4]. In order to achieve a security level equivalent to RSA with key size 1024 bits, the size of the public system parameters is 1024 bits, the size of an encrypted symmetric key is 2288 bits and the size of a re-encryption key is 832 bits. Consider now the case in which the owner stores all re-encryption keys in the proxy. Suppose that RSA public key cryptography was used. The secure publisher proxy could have been implemented by having the owner encrypt *every* symmetric encryption key with the public keys of *all* subscribers. Therefore, if 10 items had to be shared with 10 subscribers, the owner would have to generate 100 different ciphertexts, whereas our solution requires 10 re-encryption keys and 10 IBE ciphertexts. We now examine the case where an owner is online and does not store re-encryption keys in the proxy. Suppose a subscriber wants to access 10 files from the same owner. Following the RSA approach, the proxy would have to communicate 10 times with the owner in order to obtain the encryptions of the symmetric keys. With our solution, the proxy has to communicate only once, since the same re-encryption key can be used for all subsequent requests.

B. *Multi-source and multi-interface video streaming*

In this section we consider a mobile video streaming application and show how subscription proxies and the multi-source/interface features of the I-CAN architecture can be used to offload cellular traffic to Wi-Fi, offering equal or better QoE to end-users. This section is based on our previous work in [5]; the main enhancement of the current design is to exploit device-to-device communication with Wi-Fi direct, utilizing the decentralized and fast rendezvous mechanisms discussed in Section III.E. In related work, [6] investigates cooperation between mobile devices to exploit device-to-device communication for video streaming, [7] investigates adaptive video streaming over Content-Centric Networks (CNN), and [8] investigates multi-source video streaming. Our work differs in that we consider multi-source video streaming that utilizes proactive caching based on mobility and throughput prediction.

1) *Testbed implementation*

Our testbed consists of mobile devices (subscribers) that run a multi-source video streaming client. The client can utilize both cellular and Wi-Fi interfaces and request different parts (chunks) of a video from different sources, which can be publishers, subscription proxies with caches, or neighboring mobile devices. The streaming client implements the following three procedures (see [5] for details):

- Load balancing: The client adjusts the number of video chunks that it requests from each source based on the measured throughput.
- Fault tolerance: The client can detect when a source or the path from a source is down, and request video chunks from another available source.
- Prefetching: The client exploits mobility and throughput prediction to request that video chunks are prefetched by proxies at hotspots it will encounter.

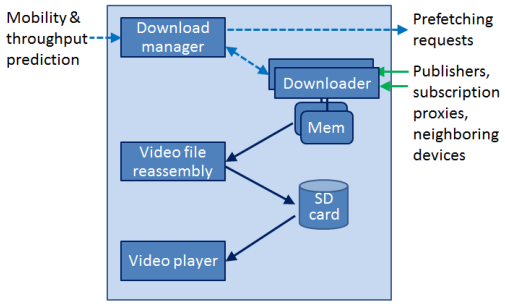

Fig. 3. Multi-source mobile video streaming client design

The high-level design of the mobile video streaming client is shown in Fig. 3. The main components are the download manager and the downloaders. The download manager uses mobility and throughput prediction information to instruct subscription proxies to prefetch and cache video chunks. The download manager also controls and synchronizes the downloaders. Each downloader transfers video chunks from a different source. The video streaming client has been tested on smartphones running Android 4.0.4 and 4.2.2.

Our testbed includes laptops with VirtualBox 4.3.6/18 and Ubuntu 13.10/10.04 virtual machines, which run publisher and subscription proxies. A device's mobility, in terms of different connectivity options and download rates for cellular, Wi-Fi, and ADSL links, is emulated based on scenarios defined in an

XML file; the XML file is downloaded by the streaming client in the beginning of each experiment; see [5] for details.

2) Experiments

We consider the scenario in Fig. 4. The access points are connected to the publisher through an emulated ADSL connection with throughput 3 Mbps. The subscriber encounters a Wi-Fi hotspot at times 0,100,200,400,500 seconds and at 300 seconds connects using Wi-Fi direct with another smartphone that has the requested video; it remains in the hotspot and smartphone range for 20 seconds. The mobility path is known to the subscriber, which instructs proxies to prefetch video chunks. The video stream has an average bit rate 1.65 Mbps and total size 120 MB. The results presented below are the average of 2-5 runs. Results for different scenarios and for the load balancing and fault tolerance mechanisms are contained in [5].

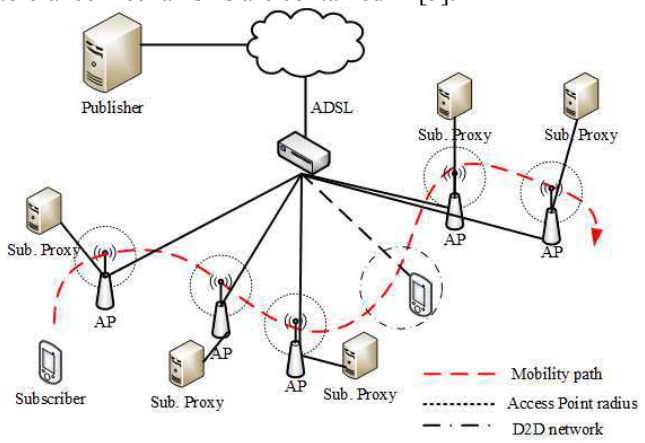

Fig. 4. Multi-source mobile video streaming scenario

Traffic offloading: In this experiment the maximum cellular rate is 2 Mbps. Fig. 1 shows the percentage of video traffic offloaded as a function of the Wi-Fi throughput, for three schemes: 1) no prefetching with the cellular network used at maximum rate, 2) prefetching with the cellular network used at maximum rate, and 3) prefetching with the cellular throughput lower than the maximum, but enough to avoid frame pauses. The results show that the percentage of offloaded traffic with prefetching increases when the Wi-Fi throughput increases; on the other hand, without prefetching the percentage of offloading is independent of the Wi-Fi throughput, since the ADSL backhaul is the bottleneck.

Video streaming QoE: The Wi-Fi throughput is set to 5 Mbps and we reduce the cellular throughput so that the subscriber experiences frame pauses. Fig. 6 shows the number of pauses as a function of the cellular throughput. The gain with prefetching is higher when the cellular throughput is smaller, which is when the higher Wi-Fi throughput can be utilized with prefetching to download video chunks faster and avoid frame pauses.

V. CONCLUSIONS

We have described the Information-Centric Access Network (I-CAN) architecture, identifying how it accounts for specific characteristics of mobile and wireless access networks. We also presented experiments with two application scenarios involving secure publisher proxies and multi-source mobile video streaming, which highlight I-CAN's key features. Ongoing work is extending the video client to perform adaptive streaming and conducting experiments with video sharing among devices using Wi-Fi direct.

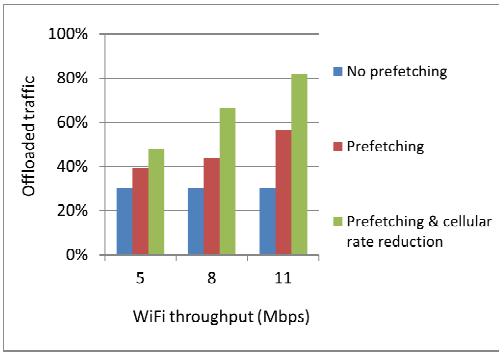

Fig. 5. Video traffic offloading

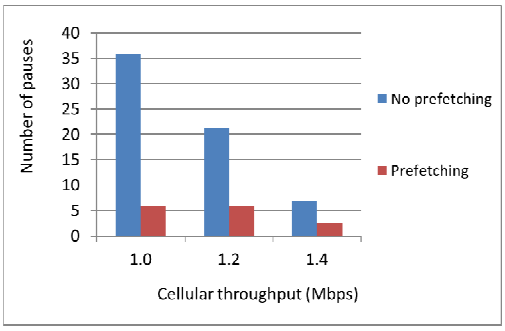

Fig. 6. Video QoE

REFERENCES

[1] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V.A. Siris, and G.C. Polyzos, "Caching and mobility support in a publish-subscribe internet architecture," IEEE Comm. Mag., vol. 50, no. 7, pp. 52-58, July 2012.

[2] X. Zhang, K. Chang, H. Xiong, Y. Wen, G. Shi, W. Guoqiang, "Towards name-based trust and security for content-centric network," Proc. 19th IEEE Int'l Conference on Network Protocols (ICNP), 2011.

[3] M. Green, G. Ateniese, "Identity-Based Proxy Re-encryption," In Katz, J., Yung, M. (eds.) Applied Cryptography and Network Security, Lecture Notes in Computer Science, vol. 4521, pp. 288-306, 2007.

[4] J. A. Akinyele, M. D. Green, and A. D. Rubin, "Charm: A framework for rapidly prototyping cryptosystems," Springer Journal of Cryptographic Engineering, vol.3, no. 2, pp. 111-128, 2013.

[5] D. Dimopoulos, Ch. Boursinos, and V.A. Siris, "Multi-Source Mobile Video Streaming: Load Balancing, Fault Tolerance, and Offloading with Prefetching," Proc. 9th Int'l Conf. on Testbeds and Research Infrastructures for the Development of Networks & Communities (TRIDENTCOM), 2014.

[6] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, "MicroCast: cooperative video streaming on smartphones," Proc. ACM MobiSys 2012.

[7] S. Lederer, C. Müller, B. Rainer, C. Timmerer, and H. Hellwagner, "Adaptive Streaming over Content Centric Networks in Mobile Networks using Multiple Links," Proc. Immersive & Interactive Multimedia Communications over the Future Internet, IEEE ICC, 2013.

[8] S.-B. Lee, A.F. Smeaton, and G.-M. Muntean, "Quality-Oriented Multiple-Source Multimedia Delivery over Heterogeneous Wireless Networks," IEEE Trans. Broadcasting, vol. 57, no. 2, pp. 216-230, 2011.