Telecommun Syst (2013) 53:299–314
DOI 10.1007/s11235-013-9699-9

Handling mobility in future publish-subscribe information-centric networks

Nikos Fotiou · Konstantinos Katsaros · George C. Polyzos · Mikko Särelä · Dirk Trossen · George Xylomenos

Published online: 20 July 2013

Abstract Future information-oriented Internet architectures are expected to effectively support mobility. PSIRP, an EU FP7 research project, designed, prototyped, and investigated a clean-slate architecture for the future Internet based on the publish-subscribe paradigm. PURSUIT, another EU FP7 research project, is further developing this architecture, which we refer to as Ψ, the Publish Subscribe Internet (PSI) architecture, extending it in various directions, including a deeper investigation of higher (transport and application) and lower layers (e.g., various link technologies, such as wireless and optical). In this paper we present the basics of the Ψ architecture, including the built-in multicast and caching mechanisms, with particular focus on mobility support. We discuss how the native, clean-slate, Ψ instantiation of the information-centric model can support mobility and also present an overlay variant of Ψ we have developed in order to provide an evolutionary path to adoption. Based on analysis and simulation we demonstrate the advantages of the proposed architecture compared to well established solutions such as Mobile IPv6.

Keywords Future internet · Content centric networks · Mobile IP

N. Fotiou (✉) · K. Katsaros · G.C. Polyzos · G. Xylomenos
Mobile Multimedia Laboratory, Athens University of Economics and Business, Athens 104 34, Greece
e-mail: fotiou@aueb.gr

K. Katsaros
e-mail: ntinos@aueb.gr

G.C. Polyzos
e-mail: polyzos@aueb.gr

G. Xylomenos
e-mail: xgeorge@aueb.gr

D. Trossen
Computer Laboratory, University of Cambridge, William Gates Building, 15 JJ Thomson Avenue, Rm FE13 Cambridge, UK
e-mail: dirk.trossen@cl.cam.ac.uk

M. Särelä
Department of Communications and Networking, Aalto University, Otakaari 5, 02150 Espoo, Finland
e-mail: mikko.sarela@aalto.fi

1 Introduction

The current Internet architecture remains relatively unchanged since its inception. Nevertheless as the Internet evolves—along with users' needs and requirements—it seems incapable of overcoming various challenges, including security, mobility, scalability, quality of service, and economics [9]. A key goal of the Internet was to efficiently interconnect mainframes and minicomputers and to provide efficient remote access to them. This end-to-end approach and especially its specific practical implementation, however, have been identified as the root cause of many limitations of the current Internet architecture. Various add-ons, such as NATs, Mobile IP, CDNs, peer-to-peer (P2P) overlays, etc., all violate, in various ways, several aspects of the original Internet architecture in order to provide features that were not part of the original design (or the original requirements). Moreover, the original Internet architecture and protocols were developed assuming a benign and cooperative environment, which is far from today's reality, where competition is widespread and lack of trust and security threats, such as malware, denial of service attacks, and phishing, have become more and more prevalent.

It seems that it is time for a shift from the current Internet, which interconnects machines, towards a new Internet that

interconnects information. PSIRP (Publish-Subscribe Internet Routing Paradigm), an EU FP7 funded research effort, has created, implemented, and initially evaluated a clean-slate, information oriented future Internet architecture; we call it the *Publish-Subscribe Internet* (PSI) architecture, or Ψ for short. This architecture aims at overcoming most limitations of the current Internet and at emphasizing the role of information as the main building block of the (future) Internet. This new architecture is based on a paradigm completely different from the current one. Ψ is based on pure, through-the-stack application of the Publish-Subscribe paradigm. The Ψ architecture, was produced with specific proposals in many key areas, such as rendezvous, topology formation and routing, and information forwarding. It includes integrated support for anycast and multicast, caching, multihoming and mobility, as well as security and privacy. A prototype of this architecture has been implemented and is available under open source license terms.[1] Moreover, a Ψ testbed across Europe is being established for testing native Ψ applications.

The current Internet architecture has been designed in a way that does not facilitate mobility. Its end-host centric nature poses barriers which various add-on protocols—such as Mobile IP—strive to overcome. The location dependent end-host identification—using the IP protocol—is regarded as the root cause of the mobility problems that the Internet faces. By taking into consideration the continuously increasing number of lightweight computers and smart phones, that can be always connected, it can be foreseen that the problem of mobility will be of bigger importance in the near future.

The purpose of this paper is to introduce the basics of the Ψ architecture with a specific focus on mobility support. In particular, we discuss how mobility can be handled by future publish-subscribe information-centric networks, using as a reference point the Ψ architecture. We investigate the various high-level mobility support functions and how the Ψ architecture facilitates them. We also use an overlay Ψ based simulation to evaluate the performance of the solution at the intra-domain level and compare it with traditional Internet mobility solutions.

The remainder of this paper is organized as follows. Section 2 presents related work in similar research efforts. Section 3 introduces the Ψ architecture, its core components and functions as well as the so-called 'bubbles' layering model. Section 4 analyzes mobility in Ψ and how various types of mobility are handled in the Ψ architecture. In Sect. 5 we present Ψ's overlay variant, which we then evaluate in an environment with mobile nodes in Sect. 6. Finally, in Sect. 7, our conclusions as well as thoughts for additional work are presented.

[1] http://www.fp7-pursuit.eu/PursuitWeb/?page_id=338.

2 Related work

A substantial body of research has been devoted to mobility support in IP networks, yielding a variety of approaches, spanning the TCP/IP protocol stack. The Mobile IP (MIP) protocol [18], along with its enhancements (Hierarchical MIPv6 [34]—which handles micro-mobility [13], Fast Handover for MIPv6 scheme [24] and Proxy Mobile IPv6 [14]), aims at solving the problem at the network layer, by propagating the changes of a Mobile Node's (MN's) network address either to a gateway at the MN's home network or the node(s) currently communicating with the MN, i.e. the Correspondent Nodes (CNs). The Host Identity Protocol (HIP) [27] introduces an additional layer between the IP and the Transport layers. The purpose of this layer is to decouple identity from location enabling the maintenance of active sessions during handoffs. The Session Initiation Protocol (SIP) [30] has also been proposed for the support of mobility at the application layer [33]. In all these approaches, the target is to propagate the network address changes to appropriate nodes in the network (usually the CNs). However, this approach only provides an "add-on" solution that actually attempts to circumvent the problem rather than solve it. Multicast assisted mobility, has also been studied in the context of IP multicast (e.g., [10]). However, these approaches suffer from the limited deployment of IP multicast [7].

Mobility in overlay publish-subscribe architectures has been widely studied. Fiege et al. [11] modified the REBECA publish-subscribe middleware in order to support physical mobility of the nodes as well as *location-dependent subscriptions*. Huang et al. [15] showed how the anonymity, the dynamism, the asynchrony, as well as the multicast nature, of publish/subscribe architectures make them ideal for wireless and mobile environments. Moreover they discussed how various publish/subscribe models can be adapted to a mobile environment. Muthusamy et al. [28] studied the mobility of publishers in publish-subscribe architectures and they developed an algorithm for handling it.

Mobility has also been studied in various content or other information-centric architectures that bear similarities with Ψ. i3 [35] advocates indirection as the solution to the problems that point-to-point (i.e., end-point oriented) communication poses to mobility. i3 implements an IP overlay network that replaces the point-to-point communication model with a rendezvous-based paradigm, where senders send packets to a specific rendezvous-point while receivers issue triggers on specific packet identifiers. i3 is a mobility friendly architecture and it has been found to offer location privacy and to perform generally better than MIP when it comes to stretch and to fault tolerance [40]. Ψ uses similar concepts through the rendezvous and topology formation processes.

Routing on Flat Labels (ROFL) [3] and the Data Oriented Network Architecture (DONA) [25] propose using flat information identifiers rather than hierarchical, location-based identifiers, and are concerned, among other things, with mobility. DONA proposes a new identification scheme based on flat, self-certifying identifiers, that enables 'finding' content, as a replacement to the DNS naming resolution scheme. The 'finding' mechanisms proposed by DONA facilitate the deployment of caches, the establishment of multicast delivery trees and enable locating the nearest copy of the content by employing anycast. Nevertheless DONA—in contrast to Ψ—does not propose any new technique for transferring data, therefore it is only concerned with the mobility of the node that hosts the content. ROFL investigates the possibility of having an internetworking architecture solely based on flat identifiers, using hierarchical DHTs. Nodes in ROFL are identified by location independent labels and, therefore, mobility is facilitated. The overall architecture distinguishes between stable hosts and ephemeral hosts, which includes mobile nodes. Ephemeral hosts are treated in a different way than stable hosts; their participation to the overall architecture and operation is the minimum possible, so that their mobility will not significantly impact the performance of the system. Ψ does not distinguish between stable and mobile nodes, as by design mobility is considered a universal property and a typical condition.

CCNx [5] is another research project aiming at redesigning the Internet with a content-centric perspective. In the proposed architecture, in which routing is based on hierarchical naming, consumers ask for content by broadcasting 'Interest' packets that contain the name of the content requested. Any 'Data' packet whose content name is a suffix of the name in the 'Interest' packet is assumed to satisfy this interest. CCNx utilizes Listen First Broadcast Later protocol (LFBL) [26] to implement mobility. This protocol enables the requesting node to pick the preferred data source for receiving the desired content. Moreover a potential forwarder, after receiving a packet, listens to the channel in order to discover if a more suitable node has already forwarded this packet. Otherwise it forwards the packet itself towards the packet destination. CCNx facilitates mobility by supporting caching, by enabling nodes to transmit 'Interest' packets simultaneously from multiple interfaces and by having its transport designed to operate on top of unreliable packet delivery services [17]. Nevertheless the absence of the notion of the rendezvous point, the unstructured organization of the information and the advertisement of content through flooding will possibly have some negative effects when it comes to supporting mobility. Table 1 presents a high-level comparison of the major information-centric architectures with respect to mobility support.

3 Overview of the Ψ architecture

The Ψ architecture abides by the publish-subscribe paradigm. Such architectures are mainly composed of three entities; the publishers, the subscribers and an event notification service [8]. Publishers are information providers who advertise the availability of specific pieces of information by issuing *publication* messages. Subscribers on the other hand are information consumers, who express their interest in specific information items by issuing *subscription* messages. The event notification service is responsible for locating the publishers who provide the information items that satisfy the consumers' subscriptions and for initiating a forwarding process from the information providers towards the

Table 1 Overview of related work

	Type of mobility	End-to-end delay	Caching and multicast
MIPv6	Hierarchical MIPv6 flavor distinguishes local from global (host) mobility, based on predefined anchor points. No support of vertical handoffs	Communication with the CN or predefined anchor points may incures high delays	Not supported
i3	Host, personal and session mobility	Multicast, caching and rendezvous in favor of small delays	Supported by the overlay
ROFL	Host mobility	Rendezvous functionality and isolation of mobile nodes in favor of small delays	Supported by in-network mechanisms
DONA	Host mobility	Rendezvous functionality in favor of small delays	Supported by in-network mechanisms
CCNx	Host mobility	Everytime the mobile node changes position it has to flood its request, therefore end-to-end delay may be high	Supported by in-network mechanisms
Ψ	Local vs. global (host mobility): based on dynamically defined anchor points (multicast forwarding). Dynamic vs. static: supports vertical handoffs	Multicast, caching, fast authentication and multi level rendezvous in favor of small delays	Supported by in-network mechanisms

information consumers. The publication and the subscription operations are decoupled in time, i.e., they do not have to be synchronized. Moreover publishers do not have to be fully aware of the subscribers and vice versa.

Publishers and subscribers are the principal actors of the Ψ architecture. An event notification service, which is referred to as the rendezvous network, is responsible for matching subscriptions with the appropriate publications as well as for initiating the information forwarding process from publisher towards subscriber(s). The rendezvous network is composed of several rendezvous nodes (RNs), each of which is responsible for a set of publications. The RN that is responsible for a publication is known as the Rendezvous Point (RP) of this publication. The existence of RPs ensures a balance of power between sender and receiver, i.e., no information is delivered without explicit signaling of availability (publish) and interest in it (subscribe).

The Ψ architecture regards information as its building block; everything is information and information is everything [36]. Information is organized in a hierarchical way, so small 'meaningless' pieces of data, which can be of arbitrary size, are combined into large complex information items—such as files, documents pictures and videos. Every piece of information is identified with a (statistically) unique label. This label actually denotes the RP in which subscriptions and publications regarding this information item will be matched; for this reason this label is referred to as the *rendezvous identifier* (RId)

Scoping mechanisms are used to limit the reachability of the information to the parties having access to that particular *scope* [20]. Within a scope the architecture is neutral with regard to the semantics and structure of the data, although governance rules regarding the available information may be defined. Scopes have a hierarchical structure where parent-children and sibling relationships exist. In Ψ, there can exist physical scopes, e.g., a corporate network, and logical scopes, e.g., a social network (hierarchy). Scopes are identified by a subclass of the rendezvous identifiers, the *scope identifier* (SId). The SId denotes the specific scope within which the information is reachable. RIds and SIds are independent from the endpoints producing and consuming the associated information items. Flat and endpoint independent labels seem to be a natural choice for information oriented architectures as they clearly separate location from identity, allowing for properly incorporating mobility, multicasting, and multihoming into the architecture, as well as a more comprehensive notion of identity [3].

Every piece of information is attached to at least one specific scope, which is represented by the scope identifier that publishers set when they publish information. Several mechanisms are used to control the scope of a piece of information. These mechanisms include access control, DRM, user authentication, and many others. Information items may be

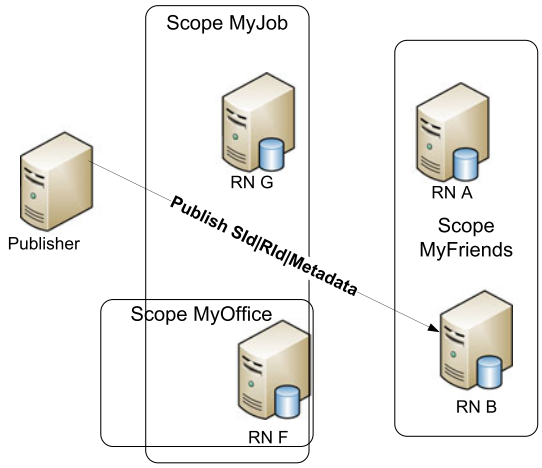

Fig. 1 Publication in Ψ

part of multiple scopes. For example, an information item (such as an image) may belong to a specific university scope (which a student attends) and at the same time it may belong to a specific family scope (that of the student's family) allowing the student to efficiently share her information only with the people related with her university and her family! Each scope is managed by at least one RN.

Whenever publishers wish to issue a new publication, they have to use two identifiers: RId and SId. A publication's RId can be derived by an application specific function. A publication's SId should denote to which extent the publisher wishes the publication to be made available. Prior to publishing an information element, publishers have to locate the nodes that are responsible for managing the desired scope. One of these nodes will be the RP for the publication. What is actually published to the RP is the publication's metadata, which contain information specific to the actual publication; this can be for instance the author of the publication, its size and perhaps a small description of it.

Figure 1 depicts the publication process in Ψ. In this figure there exist three scopes; the scope *MyJob*, the scope *MyOffice* and the scope *MyFriends*. The scope *MyOffice* is a child of the scope *MyJob*. The scopes *MyJob* and *MyOffice* are managed by *RN G* and *RN F* respectively whereas the scope *MyFriends* is managed by two RNs; *RN A* and *RN B*. In this example, the publisher has created a publication that is intended for the scope *MyFriends*, therefore he chooses between *RN A* and *RN B* which one is going to be the RP for this publication. As *RN B* is responsible for managing the RId chosen for this publication, it becomes its RP.

In order for a subscriber to access a publication she must be aware of its RId and SId. These two identifiers can be discovered using various application layer mechanisms, e.g., they can be preconfigured for some type of applications or protocols, or they can be learned using a search engine. The subscriber expresses her interest in a specific publication by

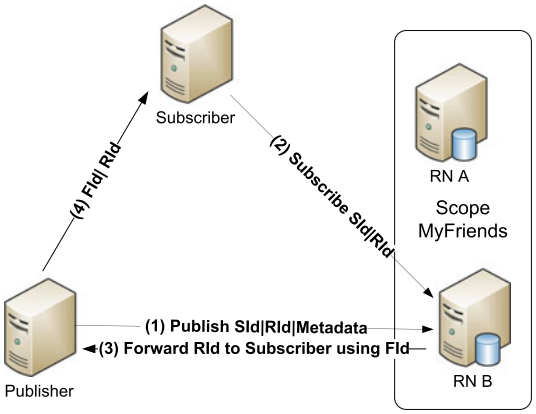

Fig. 2 Subscription in Ψ. The numbers within the parenthesis denote the messages' sequence

issuing a subscription message towards the publication's RP. Upon receiving a subscription message, and provided that there exists a publication that matches it, the RP initiates a process that creates a forwarding path from the publisher towards the subscriber. This forwarding path is identified by a *forwarding identifier* (FId). The FId is a bloom filter based structure—known as zFilter [19]—that includes all the link identifiers which the publication needs to traverse in order to reach the subscriber(s). Multicast is the preferred delivery method in Ψ, therefore in cases where there exist multiple subscribers for a particular RId, a multicast tree is created. Moreover caches may act as publishers and the RP can choose a cache-publisher that is closer to the subscriber than the original publisher. Figure 2 illustrates a subscription example. During step (1) a publisher issues a publication to the scope *MyFriends*. *RN B* becomes the RP for this publication. Later on a subscriber interested in this publication sends a subscription message to *RN B* (2). *RN B* initiates a process that creates a path from the publisher towards the subscriber and informs the publisher of the FId of this path (3). Finally the publisher forwards the publication to the subscriber through this path (4).

In order to accomplish these operations the Ψ architecture implements recursively three functions; the *Rendezvous*, the *Topology* and the *Forwarding* (RTF) functions. Each recursive step takes place in its own layer, called *bubble* [12]. Each bubble implements scope-specific RTF functions to enable the provisioning of the information within a scope. The rendezvous function is responsible for matching subscribers' interests with publications. The topology function monitors the network topology and detects changes using various techniques, depending on the bubble it is implemented in. The topology function is also responsible for creating information delivery paths at different levels of the inter-domain system. Finally, the forwarding function implements information forwarding through the delivery paths.

Figure 3 shows a particular instance of the bubble model. An information item from the shown OS in LAN 1 can traverse the following bubbles in order to be delivered/accessed by a user outside the AS 1 bubble: OS-AP-Router-LAN 1-LAN 2-LAN 3. The particular implementation of the RTF functions depends on the specific context in which the bubble has been created. For instance, methods applying locality could be utilized for the Access Point (AP) bubble, having simple forms of rendezvous, largely driven by the local attachment and by virtue of the local link discovery (i.e., literally the L2 discovery of the channel). The topology function running in the Operating System (OS) bubble is responsible for maintaining connectivity and for predefining (forwarding) labels that will be used by the forwarding function in order to forward information items through the various interfaces. Larger bubbles, such as the global one, need to solve more complex 'matching' and topology problems, which leads to more complex solutions in these areas.

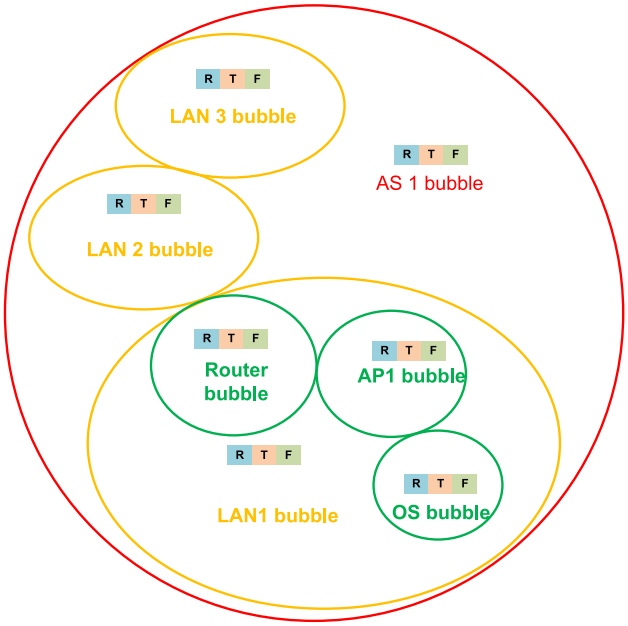

Fig. 3 Ψ's bubbles model

The implementation details of the RTF function of each bubble are invisible to the rest of the architecture. Each bubble is considered as a black box, which implements correctly the necessary functionality, therefore it should be easy to maintain, extend and modify the mechanisms deployed in a particular bubble. For the time being, various implementation alternatives are being explored and evaluated. Each function operates based on the information available to its bubble level, e.g., the Topology function of the AS 1 bubble knows that there are 3 bubbles (LAN 1, 2 and 3) that form a connected graph (LAN 3 is connected to LAN 2 and LAN 2 is connected to LAN 1), it is not aware of any topology information of the lower level bubbles.

As for the organization of bubbles, they can be included in each other or can just touch each other (implementing a sequence of domain traversals). Information within

each bubble traverses through the bubble from points on its membrane—the traversal is implemented through the proper RTF functions. The points on the membrane constitute publishers and/or subscribers of information within the enclosed but also the enclosing or touching bubble.

4 Mobility in Ψ

User mobility in Ψ is regarded as a two-dimensional issue. The first dimension concerns the scale of mobility, which can be local or global. The second dimension of the problem reflects how mobility is handled by the architecture. Mobility can be handled either in a static or in a dynamic way. When static mobility is involved, bubbles simply 'fly around' within the perimeter of their containing bubble detaching and attaching to other bubbles. In the case of dynamic mobility, temporary bubbles are created that allow for information transition between different environments—which can be different wireless technologies or even different administrative domains. Table 2 gives an overview of mobility categorization in Ψ. A user moving around with his laptop inside a campus covered by a (multi-AP) WLAN is an example of static-local mobility, whereas a Vehicular Network which involves sensing from various sensors deployed along the road directly or receiving the same information indirectly through communication with other cars is a typical example of dynamic-local mobility. On the other hand a user traveling with his mobile phone switched on in a highway, changing cells within the same operator network is a static-global case of mobility, while vertical handover without roaming is a case of dynamic-global mobility.

Figure 4 demonstrates a static-local mobility scenario. A mobile node (MN) is moving around within the premises of *LAN 1*. Its initial AP is *AP1*, and it moves to *AP3* through *AP2*. This MN is receiving a publication from a Corespondent Node (CN) located in *LAN 3*'s bubble through *LAN 2*. When the MN changes AP the information delivery tree is simply re-built by the *Router bubble* of *LAN 1*, which forwards information to the appropriate AP. In order to achieve information delivery tree reconstruction, the MN needs to issue a new subscription message from the new location. This subscription message will ultimately trigger the *Topology* and *Forwarding* functions of the *Router bubble* as well as that of the new *AP bubble*, leading to publication redirection. Outside *LAN 1's* bubble the mobility of the MN is transparent as it is 'isolated' inside LAN 1's bubble, causing no change to any other bubble in the network.

Table 2 Mobility categorization in Ψ through examples; static refers to no change in bubble; dynamic refers to the case where a new bubble is created

	Local	Global
Static	Handover in managed WLAN environment	Cell change in a mobile network
Dynamic	Nets between cars	Handover without roaming

A dynamic-global mobility scenario is depicted in Fig. 5. This scenario involves a MN that has a 3G and a WLAN interface. Initially the MN receives a publication from a CN located in *Provider's 1* bubble through its 3G interface. It then detects an available WLAN and it decides to perform a *vertical handover*. Although this scenario involves little—or no—physical MN movement, it may cause global publication-flow shift, as the 3G operator and WLAN provider can be different. The new location of the MN needs to be informed about the upcoming arrival and state needs to be transfered from the *CELL2* bubble to the *LAN 1* bubble. In order for this state transfer to occur the MN needs to inform CELL2 about its movement. CELL2 in return creates a dynamic bubble between its bubble and the *AP1* bubble. This 'dynamically created bubble' enables state transfer from the *3G* bubble to the AP to which the MN is going to be attached. The bubble of the AP in return is going to perform all the necessary actions in order to redirect the publication to the new location and when it is ready, it will inform the CELL2, using the dynamic bubble. The publication redirection requires the activation of the *RTF* functions of all bubbles between the *AP1* bubble and the *Provider 3* bubble as new subscription messages need to be sent from the new location.

4.1 Handling mobility with in-network mechanisms

Handling local mobility can be quite straightforward, especially if we consider the state that needs to be kept in the edge bubbles in order to enable multicast and caching. In

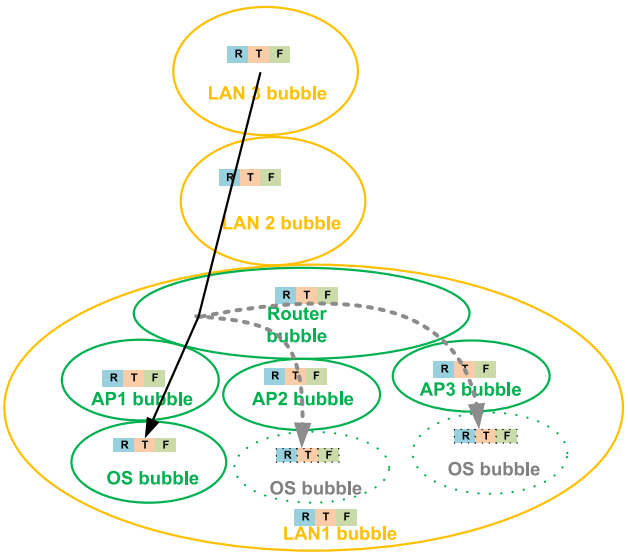

Fig. 4 Static-local mobility example in the Ψ bubbles model

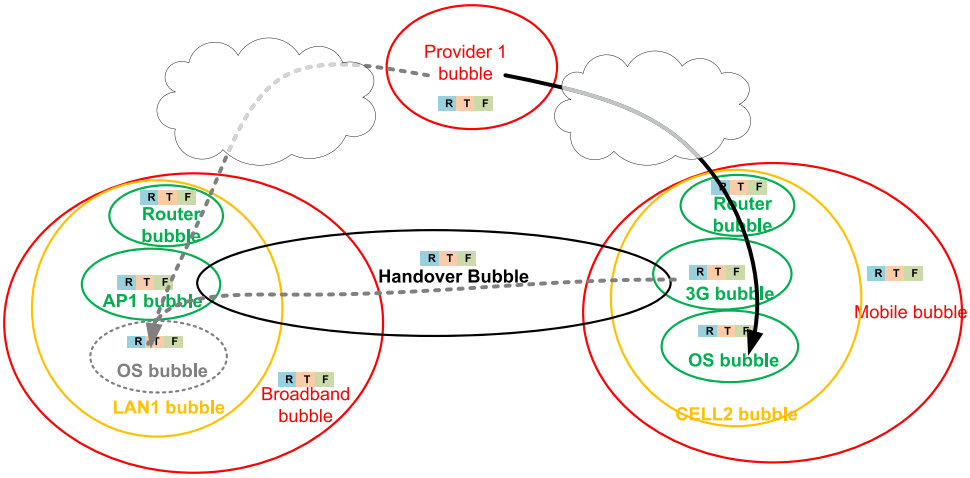

Fig. 5 Dynamic-global mobility, involving vertical handover, handled through temporary bubble creation

Fig. 4's example, the Router bubble in LAN 1 has to maintain a table that contains a list of publications and the interfaces to which each publication is forwarded. When its *Rendezvous* function receives a subscription message for a publication that already exists in its state table, the publication flow needs only to be copied (or transfered) to a new interface.

Global mobility in Ψ can be handled with in-packet Bloom filters [32]. The idea behind bloom filters is to encode the set of links comprising the delivery path or tree into a small Bloom filter, a few hundred bits long, placed in each packet. We call the filter an in-packet Bloom filter (iBF). The iBFs are collected with a signaling packet Each router names its outgoing links with a set of array positions in the iBF. The link can be added to the iBF by setting those bits using the binary OR operation. Similarly, the presence of a link can be tested by checking if the set of array positions have been set (to 1). Assuming $m = 128$-bit long string, with $k = 5$ bits set to 1, there are $\approx \frac{m!}{(m-k)!\cdot k!} \approx 3 \cdot 10^8$ different link identifiers, making link identifiers statistically unique (assuming the k set bits are randomly distributed).

The set of bit positions, called edge-pair label, can be computed at line speed based on information in the packet header such as source and destination SId address and a local secret K [19]. Computing the edge-pair labels per flow and using a cryptographically secure keyed hash makes it difficult for an attacker to guess a valid iBF for a chosen path. Assuming a maximum of 50 % of the bits set, an attacker has a $2^{-k \cdot l}$ probability of guessing an iBF for a length l path with each iBF router setting k bits. This property makes it hard for an attacker to forge iBFs.

The collected iBF is bi-directional and is used to forward payload packets between the MN and CN. As Fig. 6 shows, the iBF determines the AS-level path, while forwarding at the edges is based on the scope identifiers (SId) used in the Ψ architecture.

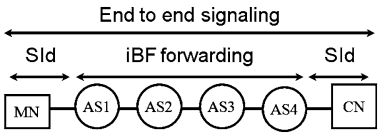

Fig. 6 Protocol messages

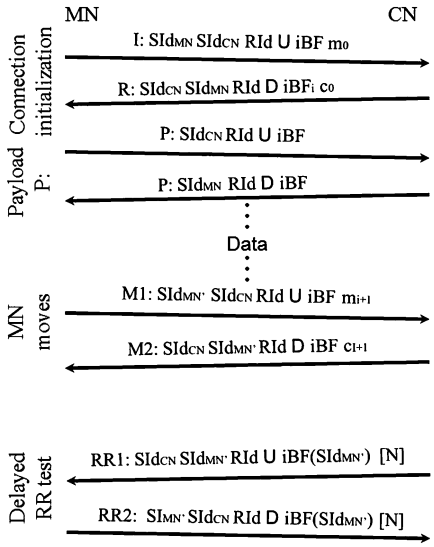

Fig. 7 Protocol messages

The protocol is shown in Fig. 7. Prior to the exchange, both the MN and CN subscribe to a well known RId that is used for publishing signaling messages. Moreover, the MN obtains an iBF to CN by feeding CN's SId to its Topology function. The MN then publishes an init (I) signaling packet which the CN will eventually receive. The packet contains a hash anchor $m_0 = h^n_{MN}(m)$ for later mobility event authentication, the SId_{CN} that identifies the CN, the RId of the publication, a pointer showing whether the iBF should be used for (U)pstream or (D)ownstream forwarding and the SId_{MN}. For scalability, ASes may partition their network

into a transit network and a set of local area networks in the edge. This way, the iBF can be used to carry the packet close to the destination. The direction bit *U/D* is used by the routers to determine the direction the Bloom filter is being used for.

The packet is forwarded through the network with the iBF. Once the packet reaches the last iBF-router, it uses a local routing protocol to forward the packet to SId_{CN}. If SId_{CN} is not in the local domain, it drops the packet. 128-bit Bloom filters carried in the packet header should be suitable for the purpose. The CN receives the packet that now contains the hash anchor and the iBF. It replies to the MN by publishing an init-reply (R) signaling packet, which contains the iBF, and its own hash chain anchor $c_0 = h_{CN}^{n}(c)$.

Later on, after the initial exchange, the CN and the MN add the iBF to each packet sent. The iBF routers determine the next-hop AS by checking which of its neighbors (combined with the incoming AS-number) edge-pair labels match the iBF. This matching is performed by the ingress border router in each AS.

When the MN moves to a new location it first requests a new iBF from the Topology function. It can then directly start publishing new payload packets to the CN. The first packet needs to contain the location update message (M1). The location update contains, among other things, the new $\text{SId}_{MN'}$, the SId_{CN}, and the next value in the hash chain $m_i + 1 = h_{MN}^{n-(i+1)}(m)$, where i is the latest used value of the hash chain. This is used to prevent man-in-the-middle attacks. The CN verifies the authenticity of the packet by verifying the revealed hash value and can directly use the iBF to publish a reply packet to the MN. The reply packet (M2) contains the next value from the h_{CN}, and the packet is forwarded using the newly collected iBF.

The delayed return routability test can be performed at any time after the initial mobility signaling, using messages RR1 and RR2. The reason for this signaling is that while the CN knows that the MN can be reached by using the iBF, it cannot be certain that the MN can be reached using $\text{SId}_{MN'}$. Should the CN move, it needs to acquire a new iBF using this SId—hence it needs to verify that the mobile node is reachable via it.

Bicasting can be used in the case of make-before-break, that is the case in which a dynamic bubble is created that connects the two providers. To do so, the mobile node signals the new location with its willingness to receive *bicast* for a time. The sender bitwise ORs the two iBFs together and sends the subsequent packet with the resulting iBF. The router at the bicast branching point automatically duplicates the packet to both destinations due to the way the iBF has been constructed. This ensures that the connection can be transferred smoothly from the old location to the new one without packet loss, or placing state maintenance requirements in the transit networks.

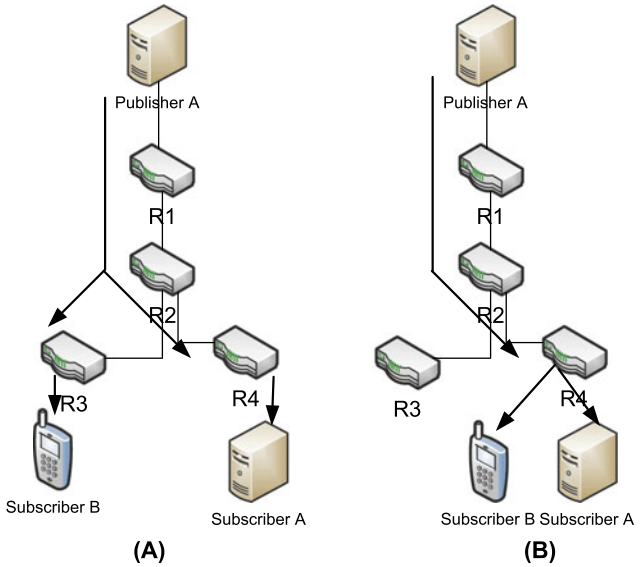

Fig. 8 Multicast assisted mobility in Ψ

4.2 Architectural design choices that facilitate mobility

The employment of multicast as the preferred delivery method, the effective use of caches and source routing based on flat location independent identifiers are three design choices of the Ψ architecture that greatly facilitate mobility.

Multicast allows for seamless hand off without interrupting the overall architecture operation. Figure 8 illustrates an example of how multicast assists mobility in Ψ. Initially two subscribers *Subscriber A* and *Subscriber B*, subscribe to the same piece of information, which in this case can be a live video stream. As can been seen in (A), a single publication flow goes from *Publisher A* to router *R2*. In *R2* this publication is split and each subscriber receives a copy of the stream. Later on *Subscriber B* decides to change location (B) and attaches itself to an access point—not shown in the figure—connected to router *R4*. From its new location *Subscriber B* issues a new subscription message for the same RId. When the rendezvous function of *R4* receives the subscription message, it already knows that there is a publication traversing it for this specific RId, therefore it only has to duplicate this publication towards the access point in which *Subscriber B* is attached.

In-network caching is another design choice of the Ψ architecture which enables negligible information forwarding resume time. An example of caching-assisted mobility is depicted in Fig. 9. In this case there exist one subscriber, *Subscriber A* and two publishers, *Publisher A* and *Publisher B*. *Publisher B* is a cache and it publishes the same publication as *Publisher A* which can be, for example, a large file such as a new Linux distribution. *Subscriber A* initially attaches to an access point—not shown in the figure—connected to

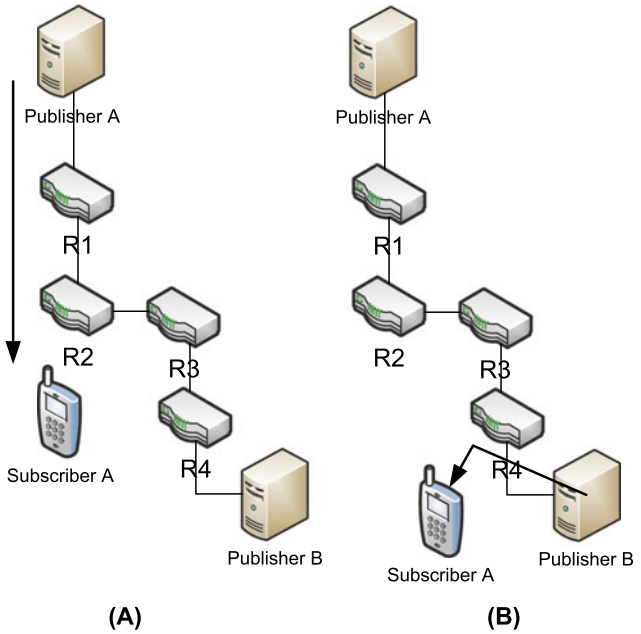

Fig. 9 Cache assisted mobility in Ψ

router *R2* and subscribes to the publication offered by *Publisher A* (and B). As a result a publication starts to flow from *Publisher A* towards *Subscriber A* through routers *R1* and *R2*. Later on *Subscriber A* changes location heading towards an access point which is connected to router *R4*. From this new location the subscriber issues a new subscription for the same publication. However now the rendezvous function of *R4* chooses as publisher *Publisher B* who is closer to *Subscriber A* and a publication starts to flow from *Publisher B* towards *Subscriber A* through *R4*. As a result *Subscriber A* notices very little delay while resuming his subscription.

In large scale mobility, multicast and caching may not be adequate. In these cases, mobility can be assisted by the source routing used in Ψ, using the zFilters to minimize latency. As described in the previous section, in most cases, when the CN receives a location update, the collected iBF and source SId suffice and it does not need to make any additional return reachability tests. This minimizes latency during hand-offs since the CN can continue sending packets to the MN after receiving a single location update. Existing protocols such as MIPv6 and HIP use 1.5 RTT to achieve the same [1].

5 An overlay implementation

In order to evaluate Ψ in an environment with local mobility, we have implemented an overlay variant of the Ψ architecture [22, 23]. Although local mobility is only one aspect of the problem, it can be argued that this aspect is the most common and with the highest potential impact on performance.

This overlay variant has been developed using the Scribe [4] publish-subscribe architecture. Scribe is based on Pastry [31], an efficient and scalable DHT substrate. Unlike other DHT schemes, Pastry attempts to employ proximity metrics, such as the number of IP hops or the round-trip time towards other nodes, when choosing among the potentially large number of DHT nodes that may relay the data in question. Due to the use of proximity metrics the average distance a message travels does not exceed 2.2 times the distance between the source and the destination in the underlying network. Pastry is used to implement the *Topology* function of the architecture, whereas Scribe implements the *Rendezvous* function. Scribe enables subscription aggregation as well as the creation of overlay multicast delivery trees. These trees are used to forward publications using TCP/IP. The (overlay) forwarding paths that are created are not optimal, therefore this overlay variant of Ψ has worse performance than a future native Ψ architecture. Even with that, the overlay variant yields better results than MIPv6 and HMIPv6, when it comes to resume time and packet loss.

In this overlay variant every router participates in both Pastry and Scribe and it can maintain Scribe multicast trees. Every router is assigned a unique ID and hence, a unique position in the identifier space. In terms of Ψ every router is a Rendezvous Node (RN) which implements the Topology and Rendezvous functions using Pastry and Scribe, respectively, whereas TCP/IP is used to implement the Forwarding function. On the other hand, (mobile) end nodes neither participate in Pastry/Scribe, nor carry an IP address. This clearly reflects our target of breaking the end-to-end semantics of today's communication. Every end node is directly connected to an access router which is called the *overlay access router* (OAR). The OAR is the router that provides access to the overlay (rendezvous) network and to the multicast communication substrate. Mobile nodes are connected to OARs through their currently associated Access Point (AP). APs act as simple bridges to the wired part of the network.

Whenever a (mobile) node wishes to act as a publisher, it sends its publication to its OAR which is then responsible to deliver it to the proper RP. In our case every publication is handled by the RN whose ID is numerically closer to the RId of the publication. Pastry is used in order to locate the proper RN. Whenever a (mobile) node wishes to subscribe to a publication, it sends a subscribe(RId) message to its OAR. When the subscribe message reaches the OAR, Scribe creates a multicast delivery tree from the publication's RP towards the subscriber(s). The procedure used is the following: When a RN receives a subscribe(RId) message, for the first time, it forwards it toward the RP. Moreover it maintains state in order to forward the publication towards the RN from which it received the subscription message (or the end host if the RN is an OAR). If the RN has already received a subscription for the specific RId, it means that it is

already part of the delivery path, therefore, it simply adds a pointer to the new RN (or end host) towards which it shall copy and forward the publication. This creates a multicast delivery tree.

What actually changes with mobility is that as mobile nodes (MNs) move from one AP to another it is possible that they change their OAR as well. In this case they must inform their new OAR about the publication they are interested in. This is accomplished by a new subscribe(RId) message that is issued towards the RP, which gets suppressed at the lowest common ancestor of the new and the old OAR in the resulting multicast tree. Of course, there is always the case that an OAR is already a member of the multicast tree for the publication the MN is interested in. This may happen because another end node that resides on that OAR has already subscribed for the same publication, or because, due to the overlay nature of Scribe, this OAR has become part of this multicast tree in order to serve another OAR as a forwarder. In that case the OAR only has to forward the publication towards the MN.

The overlay variant deviates from native Ψ in two aspects. First, the overlay variant employs a single mechanism for the Rendezvous and Forwarding functions based on the operation of Pastry and Scribe. The Rendezvous process actually creates the multicast tree that is later used to deliver the actual publication. It is noted that a DHT-like mechanism can also be employed for the implementation of the Rendezvous function in native Ψ; however this issue is outside the scope of this paper. On the forwarding plane, the overlay variant does not follow a source routing scheme as in native Ψ, since it is designed to operate on top of the underlying TCP/IP protocol stack. Second, as it focuses on local mobility support, the overlay variant follows a simpler unified approach that does not differentiate global mobility, as is the case for native Ψ. However, for local mobility support, the two mechanisms are similar, employing multicast in order to localize the routing/forwarding updates triggered by mobility.

6 Evaluation

Here we present our analytical as well as our simulation results regarding the overlay variant evaluation [21].

6.1 Signaling

We investigate the performance of the overlay variant and compare it analytically against Mobile IPv6. We chose Mobile IPv6 as it is in itself a standardized solution, and it can also serve as a means for an indirect comparison of our approach with other mobility assisting schemes. We first compare our scheme against plain Mobile IPv6 and then proceed

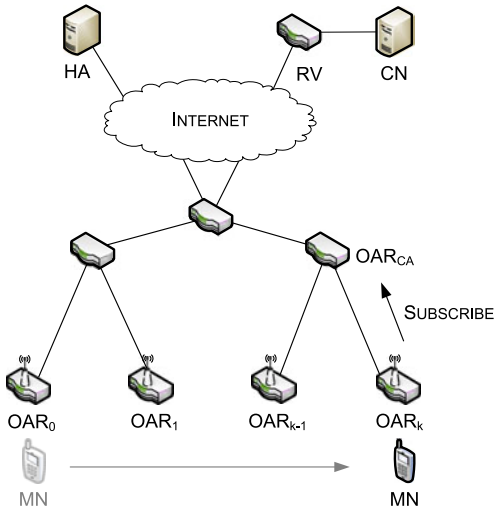

Fig. 10 Overlay handoff

with a comparison against its micro-mobility enhancement, Hierarchical Mobile IPv6. Our purpose is to demonstrate the ability of the proposed scheme to better localize routing updates, while acting as a unified solution for both macro- and micro-mobility. It must be noted however, that the presented comparisons also apply to the case of other available solutions such as the SIP protocol, which can also be used to propagate routing updates to the involved network entities, albeit at a higher layer [33].

We chose to use the handoff delay as the primary performance metric since it reflects the service disruption experienced by the user. Specifically, we consider the amount of time required for a MN to be able to resume communication after a change of network location, that is, once the MN has associated with the new wireless AP. We will refer to this metric as the *Handoff Latency* (HL). Obviously, this time is heavily affected by the signaling required for the involved network entities to be informed about the MN's change of position i.e., the time required for the routing substrate to adapt to the movement of the MN.

6.1.1 Overlay Ψ ($O\Psi$) vs. mobile IPv6

Figure 10 presents an example scenario in which a MN performs several consecutive handoffs while communicating with a corresponding node, in the case of $O\Psi$. Our target here is to investigate the impact of the signaling procedure required in the cases of Mobile IPv6 and the proposed architecture so that the MN can again become reachable after changing its point of attachment to the network.

We denote by $d_{x\to y}$ the delay of a message sent from network entity x to y. For simplicity we assume that $d_{x\to y} = d_{y\to x}$. Using the same topology for both $O\Psi$ and Mobile IPv6 (see Figs. 10 and 11), we have:

$$d_{OAR_{k-1}\to OAR_k} = d_{AR_{k-1}\to AR_k} \quad (1)$$

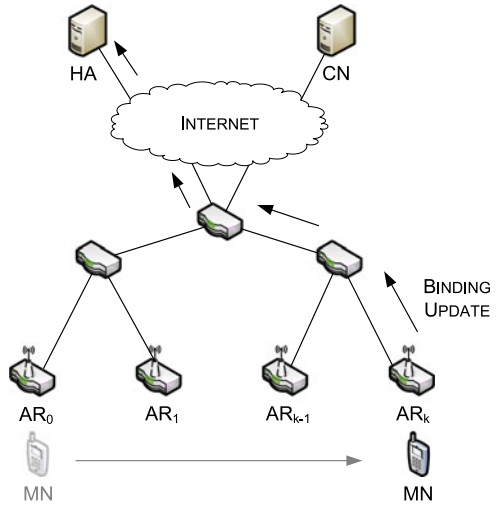

Fig. 11 Mobile IPv6 handoff (without RO)

Mobile IPv6 without RO First, we consider the case of the simple Binding Update procedure, without employing the Route Optimization (RO) procedure. Figure 11 shows the corresponding scenario under Mobile IPv6. A Binding Update (BU) message is sent towards the Home Agent (HA) of the MN and a Binding Acknowledgment (BA) message is returned to the MN.[2] Hence the HL is:

$$HL_{MIPv6} = 2 \cdot (d_{MN \to AR_k} + d_{AR_k \to HA}) \tag{2}$$

In the case of OΨ, a subscribe message is sent over the wireless medium by the MN towards the newly visited OAR, which in turn forwards it towards the Common Ancestor (CA) router of the old and the new location. The CA already receives the publication therefore it only has to forward it to the new location (this is achieved due to the multicast nature of Scribe). Considering also the acknowledgment of this packet,[3] we have:

$$HL_{O\Psi} = 2 \cdot (d_{MN \to OAR_k} + d_{OAR_k \to CA}) \tag{3}$$

The following equation expresses *Pastry*'s route convergence property.

$$d_{OAR_k \to CA} = a \times d_{OAR_{k-1} \to OAR_k}, \quad a \to 1 \tag{4}$$

Considering Eqs. (1) and (4), and assuming the delay of wireless transmissions to be the same in both scenarios, our scheme results in a smaller HL value than MIPv6 when:

$$HL_{O\Psi} < HL_{MIPv6} \Leftrightarrow a < \frac{d_{AR_k \to HA}}{d_{AR_{k-1} \to AR_k}}$$

Since according to the route convergence property $\alpha \to 1$, it is clear that our architecture results in a reduced HL compared to MIPv6, as in most cases the distance between neighboring OARs is expected to be smaller than the distance between the current OAR and the home network of the MN. Moreover, when OAR_k is already a member of the publication's multicast tree, the *HL* is expected to be further reduced i.e., $HL_{O\Psi} = 2 \cdot d_{MN \to OAR_k}$, in which case OAR_k simply acknowledges the request of the MN. This may happen because another MN attached to OAR_k has already expressed interest for the same publication, or because OAR_k is acting as a forwarding node for another OAR.

Table 3 Return Routability procedure

#	Message type	Source	Destination
1	Binding Update (BU)	MN	HA
2	Binding Ack. (BA)	HA	MN
3	Home Test Init (HoTi)	MN	HA → CN
4	Care-of Test Init (CoTi)	MN	CN
5	Home Test (HoT)	CN	HA → MN
6	Care-of Test (CoT)	CN	MN
7	Binding Update (BU)	CN	MN
8	Binding Ack. (BA)	MN	CN

Mobile IPv6 with RO Next, we consider the case of the Route Optimization (RO) procedure. In RO the Binding Update also targets the CNs the MN is actively communicating with, apart from the HA, in order to avoid triangular routing. RO is based on the Return Routability procedure, which is used to enable a CN to assure that the MN is triggering the Binding Update. Table 3 summarizes the signaling messages of the entire RO procedure. Initially, the MN issues a BU message towards its HA in order to enable the Return Routability procedure, as explained below. This message is acknowledged by the HA. Then, the MN issues two messages towards the CN, at the same time. The first message (Home Test Init, HoTi) passes through the HA of the MN, while the second (Care-of Test Init, CoTi) is send directly to the CN. As a response, the CN issues the Home Test and Care-of Test messages that follow the reverse paths of the HoTi and CoTi messages respectively. The purpose of this exchange is to allow the CN to verify that no other node but the MN has required the data to be sent towards another network address. The Return Routability procedure is completed with the reception of the HoT and CoT messages by the MN, which can then issue a BU message to the CN.

[2] We omit acknowledgment messages from the figures for clarity reasons.

[3] It is noted that the routing update takes place before the arrival of the acknowledgement packet to the MN. However, we take these messages into account for reasons of completeness and in order to provide a fair comparison of the considered protocols.

Steps 3 and 4 and steps 5 and 6 of the Return Routability procedure are considered to require very little processing and thus to take place in parallel [18]. Assuming that the HoTi and HoT messages take longer to reach there destinations compared to the CoTi and CoT messages respectively, as they pass through the HA, we can express the HL of Mobile IPv6 with RO as follows:

$$HL_{MIPv6} = 4d_{MN\to HA} + 2d_{MN\to CN} + 2d_{HA\to CN} \tag{5}$$

Based on the above, our scheme results in a smaller HL value when:

$$a < \frac{2d_{MN\to OAR_k} + d_{OAR_k\to CN} + d_{HA\to CN}}{d_{OAR_{k-1}\to OAR_k}} \tag{6}$$

Again, since $a \to 1$ and based on the expected proximity between neighboring OARs, we expect the *HL* in OΨ to be lower than in the case of Mobile IPv6 with RO. It must be noted though, that the OΨ signalling does not aim at authenticating the MN triggering the handoff, as in the case of the RO procedure. However, when consecutive handoffs are performed by a MN, the HoTi, CoTi, HoT and CoT messages are cached in the CN and the MN in order to reduce the HL. In these cases, Steps 1 through 6 are omitted, and by following the same reasoning, OΨ yields a faster handoff when:

$$a < \frac{d_{AR_k\to CN}}{d_{AR_{k-1}\to AR_k}} \tag{7}$$

6.1.2 OΨ vs. hierarchical mobile IPv6

Hierarchical Mobile IPv6 reduces the handoff delay by considering the deployment of a Mobility Anchor Point (MAP) at the gateway router of each administrative domain. The role of the MAP is then to handle all Binding Updates locally, hiding any intra-domain handoff events from the HA and the CNs. Figure 12 shows the corresponding Binding Update procedure for the considered example. In this case we have:

$$HL_{HMIPv6} = 2\cdot(d_{MN\to AR_k} + d_{AR_k\to MAP}) \tag{8}$$

Similarly with the case of regular Mobile IPv6, OΨ results in a smaller HL value when:

$$HL_{O\Psi} < HL_{HMIPv6} \Leftrightarrow a < \frac{d_{AR_k\to MAP}}{d_{AR_{k-1}\to AR_k}}$$

Again, the distance between the MAP and the current point of attachment is expected to be greater than the distance between two consecutive OARs, yielding a higher delay in the case of Hierarchical Mobile IPv6. This is because, as mentioned earlier, the MAP is located at the gateway router of the current administrative domain. Moreover, it must be stressed that although HMIPv6 succeeds in localizing mobility within administrative boundaries, it is constrained by the static character of the selected MAP points in the domain i.e., route updates always reach a predefined, MAP regardless of the location of the mobile node. On the other hand, based on the availability of multicast forwarding, OΨ requires the propagation of routing updates only to the lowest CA, allowing therefore shorter handoff periods.

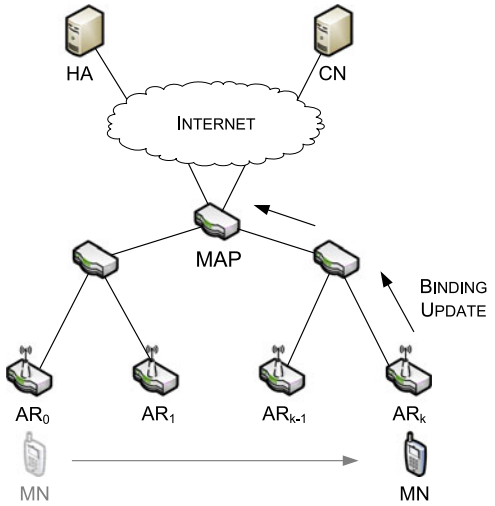

Fig. 12 Hierarchical mobile IPv6 handoff

6.2 Simulation results

To evaluate the performance of our scheme under dynamic conditions, we used the OMNeT++ simulation framework [38] enhanced with xMIPv6 [39] and OverSim [2]. In our simulations we considered a simple network topology comprising multiple OARs deployed in a grid-like topology. All OARs run the full TCP/IP protocol stack, as well as *Pastry* and *Scribe* in the case of the OΨ. All wired connections are implemented with Ethernet links. One IEEE802.11b AP is directly connected to each OAR, with neighboring APs operating in disjoint channels. We were restricted by the xMIPv6 model which does not provide an implementation of the Hierarchical Mobile IPv6 enhancement, it mandates the use of the Route Optimization procedure and also necessitates the deployment of only a single home network with a single MN in any topology.

Both the HA and the CN are attached to randomly chosen, disjoint OARs of the topology. The AP in the Home Network of a MN is directly connected to the HA rather than to the respective OAR. In the case of the proposed architecture and in order to produce comparable scenarios, we placed an extra OAR between the AP and the randomly chosen OAR acting as the initial point of attachment of the MN to the network. Figure 13 shows an example topology. The circles around each AP denote its transmission range. The

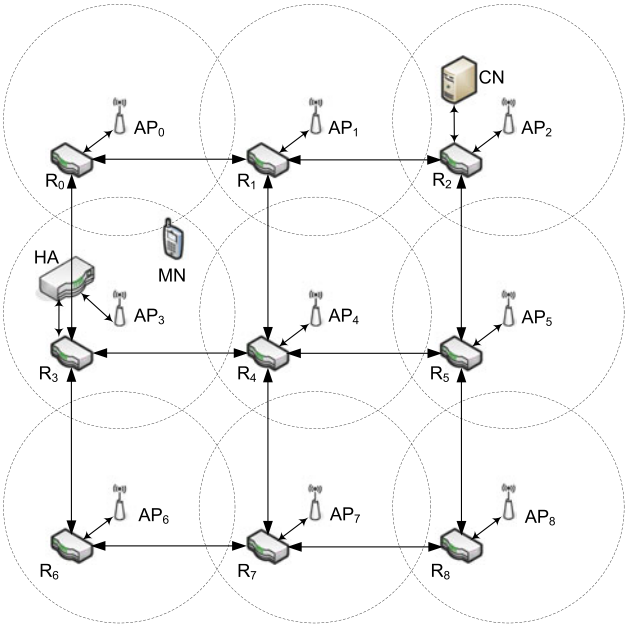

Fig. 13 Grid-like example topology

square bounded area denotes the part of the topology actually accessible by the MN. We have chosen a topology providing full wireless coverage in order to restrict the anticipated service disruption to each protocols' operation.

Our scenarios consider a single MN initially connected to its home network. Upon initialization, the MN starts to move following the mobility model described in [29]. In this model a MN moves in a straight line, makes a turn and starts over. Its speed and direction are updated every x seconds, with x following a normal distribution with a mean of 10 seconds and a standard deviation of 0.1 seconds. The speed of the MN is normally distributed with a mean of 1.39 meters/sec (approximate walking speed of 5 km/h) and a standard deviation of 0.01 meters/second. The change in direction also follows a normal distribution with an average of 0 degrees (no turn) and a standard deviation of 5 degrees. Whenever the MN reaches the limits of the simulated area it bounces with the same angle and speed.

Our measurements were based on a simple application scenario in which a stationary node sends a sequence of UDP datagrams (CBR traffic) towards the MN. The UDP stream resembles a H.264, Level 1 SQCIF video stream with 30.9 frames per second [16]. In the case of MIPv6, the CN is initially only aware of the MN's home address and uses it as the destination of its data. While moving, the MN is responsible for updating its bindings with its HA and the CN in the case of Route Optimization. In the case of the proposed architecture, the CN simply sends its UDP packets towards a rendezvous point whose position in the networks is determined by a randomly generated key and *Pastry*'s functionality. The complete set of parameter values used in our simulation environment is provided in Table 4.

Table 4 Simulation scenario parameters

Parameter	Value
Grid size	30 × 30
Number of MNs	1
Number of CNs	1
Wired connections type	100 Mbps Ethernet
Propagation delay (ms)	0.5
Data rate (Kbps)	64
Packet size (bytes)	26
Total number of packets sent	556200

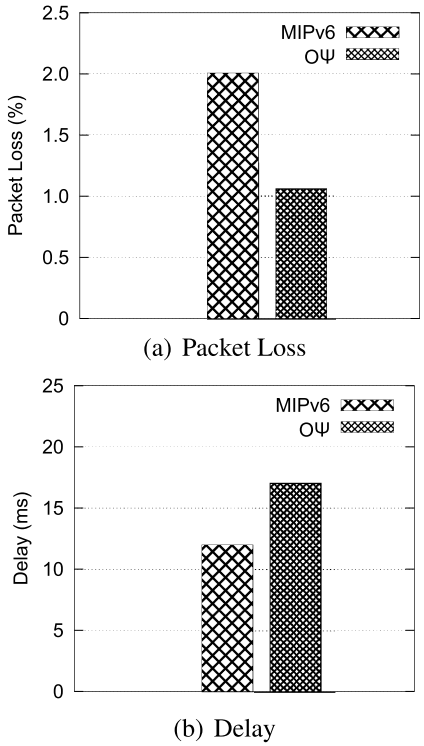

Fig. 14 OΨ vs. mobile IPv6

Figure 14 presents the results derived from the simulation scenarios described above. These results constitute the first step towards a thorough evaluation of the proposed architecture with respect to mobility.

As expected, our scheme significantly lowers packet loss in comparison to Mobile IPv6. This is justified by the localized handling of mobility. Upon each change of network position Mobile IPv6 launches the Return Routability procedure in order to apply Route Optimization. As shown in the previous section, this causes an exchange of signaling packets both with the HA and the CN. Considering the fact that these nodes may be located in distant parts of the network, this signaling may considerably delay the establishment of new routes for the traffic destined to the MN. On the other hand, in our architecture this signaling overhead is

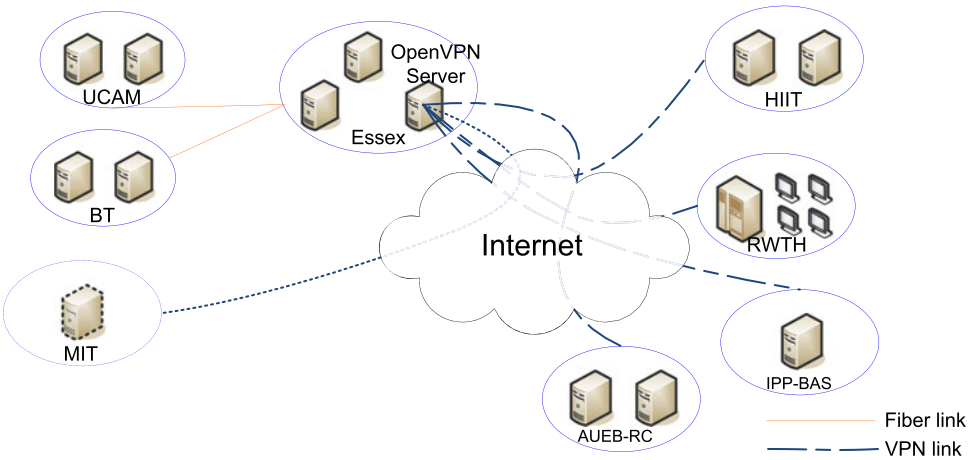

Fig. 15 Ψ's testbed topology

reduced since only the CA node in the multicast tree is notified about the MN's change of position. The significance of the reduced Resume time is depicted in the Packet loss metric where we see a noticeable difference between the two considered approaches. Furthermore, it must be noted that by localizing routing updates our architecture enables the actual delivery of packets in transit in the scenarios where the CN resides in a distant area of the network. However, these improvements come at the cost of end-to-end delay. Indeed, as we see in Fig. 14(b), our approach results in a higher end-to-end delay. As explained earlier, this is due to stretch imposed on the routing due to the reliance on a DHT substrate. We must note however, that this increase could be acceptable for non-interactive streaming applications as the ones considered in this work.

7 Conclusions and future work

A clean-slate Internet architecture based on the Publish-Subscribe paradigm is envisioned. In this architecture mobility will not be supported by add-on mechanisms as an afterthought, but it will be considered as a typical condition, which will be handled by in-network mechanisms. In this paper we provided an overview of the Ψ (Publish-Subscribe Internet) architecture, focusing on mobility support. We presented its core components and functions and we showed how its information-oriented nature, its multicast delivery, as well as its in-network caching mechanisms, all facilitate mobility support. However, the main advantages of Ψ lie in its flexibility and support for late binding when it comes to specific tussles [6] as argued by Trossen et al. [37].

Along with the native, clean-slate, architecture, an overlay variant of Ψ (which similar functionality and abides by its core principles) has been developed in order to provide an evolutionary path to adoption. Through analytical evaluation and simulation we demonstrated the efficacy of this overlay variant and we compared it to present day Internet solutions, such as Mobile IP with route optimizations and Hierarchical Mobile IP; in both cases the proposed overlay solution had considerably better performance. We used the Overlay Ψ implementation to more easily and directly compare it with traditional Internet mobility solutions.

Future work specifically in this sub-area includes further refinement and optimizations of the mechanisms for mobility support in a native Ψ environment and further evaluation through experimentation and comprehensive simulations based on realistic workloads for the new architecture (not based on current Internet architecture workloads). A Ψ testbed across Europe is being established for testing native Ψ applications (Fig. 15). This testbed is being extended in various ways, as part of the EU FP7 PURSUIT[4] research project, including wireless equipment which will allow us to perform experiments with wireless and mobile nodes.

Acknowledgements The work reported in this paper was supported in part by the FP7 ICT project PSIRP, under contract ICT-2007-216173 and by the FP7 ICT project PURSUIT, under contract ICT-2010-257217.

References

1. Aura, T., Roe, M., & Arkko, J. (2002). Security of internet location management. In *Proceedings of the 18th annual computer security applications conference* (pp. 78–87).
2. Baumgart, I., Heep, B., & Krause, S. (2007). Oversim: a flexible overlay network simulation framework. In *Proceedings of the IEEE global internet symposium* (pp. 79–84).
3. Caesar, M., Condie, T., Kannan, J., Lakshminarayanan, K., & Stoica, I. (2006). ROFL: routing on flat labels. In *Proceedings of the ACM SIGCOMM conference* (pp. 363–374). New York: ACM.
4. Castro, M., Druschel, P., Kermarrec, A., & Rowstron, A. (2002). SCRIBE: a large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communications*, *20*(8), 1489–1499.

[4] http://www.fp7-pursuit.eu/.

5. CCNx: web site (2010). http://www.ccnx.org.
6. Clark, D., Wroclawski, J., Sollins, K., & Braden, R. (2005). Tussle in cyberspace: defining tomorrow's internet. *IEEE/ACM Transactions on Networking*, *13*(3), 462–475.
7. Diot, C., Neil, B., Bryan, L., & Balensiefen, K. D. (2000). Deployment issues for the IP multicast service and architecture. *IEEE Network*, *14*, 78–88.
8. Eugster, P. T., Felber, P. A., Guerraoui, R., & Kermarrec, A. M. (2003). The many faces of publish/subscribe. *ACM Computing Surveys*, *35*(2), 114–131.
9. Feldmann, A. (2007). Internet clean-slate design: what and why? *Computer Communication Review*, *37*(3), 59–64.
10. Festag, A., Karl, H., & Wolisz, A. (2007). Investigation of multicast-based mobility support in all-IP cellular networks. *Wireless Communications and Mobile Computing*, *7*, 319–339. doi:10.1002/wcm.v7:3.
11. Fiege, L., Gartner, F., Kasten, O., & Zeidler, A. (2003). Supporting mobility in content-based publish/subscribe middleware. In M. Endler & D. Schmidt (Eds.), *Lecture notes in computer science: Vol. 2672. Proceedings of the middleware conference* (p. 998). Berlin: Springer.
12. Fotiou, N., Trossen, D., & Polyzos, G. (2012). Illustrating a publish-subscribe Internet architecture. *Telecommunications Systems*, *51*(4), 233–245. doi:10.1007/s11235-011-9432-5.
13. Frikha, M., & Maalej, L. (2006). Micro mobility in the ip networks. *Telecommunications Systems*, *31*, 337–352. doi:10.1007/s11235-006-6722-4.
14. Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., & Patil, B. (2008). *Proxy mobile IPv6*. http://www.ietf.org/rfc/rfc5213.txt.
15. Huang, Y., & Garcia-Molina, H. (2004). Publish/subscribe in a mobile environment. *Wireless Networks*, *10*, 643–652.
16. ITU: H.264 (2007). *Advanced video coding for generic audiovisual services*. http://www.itu.int/rec/T-REC-H.264.
17. Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., & Braynard, R. L. (2009). Networking named content. In *Proceedings of the ACM CoNEXT* (pp. 1–12). New York: ACM.
18. Johnson, D., Perkins, C., & Arkko, J. (2004). *Mobility Support in IPv6*. http://www.ietf.org/rfc/rfc3775.txt.
19. Jokela, P., Zahemszky, A., Esteve Rothenberg, C., Arianfar, S., & Nikander, P. (2009). Lipsin: line speed publish/subscribe internetworking. In *Proceedings of the ACM SIGCOMM conference* (pp. 195–206). New York: ACM.
20. Jokela, P. (Ed.) (2010). *PSIRP deliverable 2.2, conceptual architecture definition, component descriptions, and requirements (d2.2)*. http://www.psirp.org/
21. Katsaros, K. (2010). *An information-centric overlay network architecture for content distribution and mobility support*. Ph.D. Thesis, Athens University Of Economics and Business.
22. Katsaros, K., Fotiou, N., Polyzos, G. C., & Xylomenos, G. (2009). Overlay multicast assisted mobility for future publish/subscribe networks. In *Proceedings of the ICT mobile summit*, Santander, Spain.
23. Katsaros, K., Fotiou, N., Polyzos, G. C., Xylomenos, G., & Athens, G. (2009). Supporting mobile streaming services in future publish/subscribe networks. In *Proceedings of the wireless telecommunications symposium*, Prague, Czech Republic (pp. 337–343). New York: IEEE Press.
24. Koodli, R. (2005). *Fast handovers for mobile IPv6*. http://www.ietf.org/rfc/rfc4068.txt.
25. Koponen, T., Chawla, M., Chun, B. G., Ermolinskiy, A., Kim, K. H., Shenker, S., & Stoica, I. (2007). A data-oriented (and beyond) network architecture. In *Proceedings of the ACM SIGCOMM conference* (pp. 181–192). New York: ACM.
26. Meisel, M., Pappas, V., & Zhang, L. (2010). Ad hoc networking via named data. In *Proceedings of the fifth ACM international workshop on mobility in the evolving internet architecture, MobiArch'10* (pp. 3–8). New York: ACM.
27. Moskowitz, R., Nikander, P., Jokela, P., & Henderson, T. (2006). *Host Identity Protocol (HIP)*. http://www.ietf.org/rfc/rfc5201.txt.
28. Muthusamy, V., Petrovic, M., Gao, D., & Jacobsen, H. A. (2005). Publisher mobility in distributed publish/subscribe systems. In *Proceedings of the IEEE international conference on distributed computing systems workshops*, Ohio, USA (pp. 421–427).
29. Perkins, C., & Wang, K. Y. (1999). Optimized smooth handoffs in mobile IP. In *Proceedings of the IEEE international symposium on computers and communications* (pp. 340–346).
30. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., & Schooler, E. (2002). *SIP: Session Initiation Protocol*. http://www.ietf.org/rfc/rfc3261.txt.
31. Rowstron, A., & Druschel, P. (2001). Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Lecture notes in computer science: Vol. 2218. Proceedings of the 18th IFIP/ACM international conference on distributed systems platforms (Middleware)* (pp. 329–350). Berlin: Springer.
32. Sarela, M., Ott, J., & Ylitalo, J. (2010). Fast inter-domain mobility with in-packet bloom filters. In *Proceedings of the 5th ACM international workshop on mobility in the evolving internet architecture*, Chicago, IL, USA.
33. Schulzrinne, H., & Wedlund, E. (2000). Application-layer mobility using SIP. *ACM SIGMOBILE Mobile Computing and Communications Review*, *4*(3), 47–57. doi:10.1145/372346.372369.
34. Soliman, H., Castelluccia, C., Malki, K. E., & Bellier, L. (2005). *Hierarchical Mobile IPv6 mobility management (HMIPv6)*. http://www.ietf.org/rfc/rfc4140.txt.
35. Stoica, I., Adkins, D., Zhuang, S., Shenker, S., & Surana, S. (2004). Internet indirection infrastructure. *IEEE/ACM Transactions on Networking*, *12*(2), 205–218.
36. Tarkoma, S. (Ed.) (2010). *PSIRP deliverable 2.3, architecture definition, component descriptions, and requirements (d2.3)*. http://www.psirp.org/
37. Trossen, D., Sarela, M., & Sollins, K. (2010). Arguments for an information-centric internetworking architecture. *Computer Communication Review*, *40*(2), 26–33.
38. Varga, A. (2010). Omnet++ network simulator home page http://www.omnetpp.org.
39. Yousaf, F. Z., Bauer, C., & Wietfeld, C. (2008). An accurate and extensible mobile IPv6 (xMIPV6) simulation model for OMNeT++. In *Proceedings of the 1st international conference on simulation tools and techniques for communications, networks and systems & workshops*, ICST, Brussels, Belgium (pp. 1–8).
40. Zhuang, S., Lai, K., Stoica, I., Katz, R., & Shenker, S. (2005). Host mobility using an internet indirection infrastructure. *Wireless Networks*, *11*(6), 741–756.

Nikos Fotiou is a Ph.D. Candidate in Computer Science at AUEB. He has received his Dipl. in Information and Communication Systems Eng. from the University of the Aegean in Samos, Greece (2005) and his M.Sc. in Internetworking from the Royal Institute of Technology (KTH) in Stockholm, Sweden (2007). He has participated in many EU projects including PSIRP, PURSUIT and Euro-NF. His current research interests include publish/subscribe architectures and information oriented security mechanisms

Konstantinos Katsaros received his B.Sc. in Informatics (2003), and his M.Sc. (2005) and Ph.D. (2010) degrees in Computer Science from AUEB. His Ph.D. thesis was on information-centric networking with a particular focus on content distribution and mobility support. Since then has worked as a research associate at the Mobile Multimedia Laboratory (AUEB) and the Laboratory of Information, Networking and Communication Sciences (LINCS) at Telecom ParisTech, France. He has also worked in mobile grid computing, cognitive radio and multicast/broadcast service provision over cellular networks. He has participated in many EU projects including PSIRP and PURSUIT. His current research interests include smart grid communications, information-centric network architectures and cloud computing and networking. Currently he is a member of the Communication and Information Systems Group at the Department of Electronic and Electrical Engineering, University College London, UK.

George C. Polyzos is a Professor of Computer Science at AUEB, is leading the Mobile Multimedia Laboratory. He was Professor of Computer Science and Engineering at the University of California, San Diego, where he was co-director of the Computer Systems Laboratory, member of the Steering Committee of the UCSD Center for Wireless Communications, and Senior Fellow of the San Diego Supercomputer Center. He has received his Dipl. in EE from the National Technical University in Athens, Greece (1982) and his M.A.Sc. in EE (1985) and Ph.D. in Computer Science (1989) from the University of Toronto. His current research interests include Internet architecture and protocols, ubiquitous computing, network security, mobile multimedia communications, wireless networks, and performance analysis of computer and communications systems. He was an organizer of the EIFFEL Think Tank, on the Steering Board of the Euro-NF Network of Excellence and led his lab's participation in the EU FP7 PSIRP and PURSUIT projects that developed the PSI clean-slate ICN architecture. He has been on the Program Committees of many conferences and workshops and a reviewer for many scientific journals and research funding agencies.

Mikko Särelä works as a postdoctoral researcher at Aalto University. Prior to joining Aalto University, he did his Ph.D. working on information centric network architecture and future Internet at Nomadiclab, Ericsson Research. His current research interests include information centric networking, ethernet scalability and security, and network security.

Dirk Trossen is a Senior Researcher in the Computer Laboratory at Cambridge University, has more than ten years of experience in network architectures and wireless technology with main contributions in the area of inter-domain networking, seamless handovers and physical network overlays. He designed a platform for participatory wireless sensing, available under open source license. Dirk is one of the main drivers of the EIFFEL think tank. He is currently technical lead for the EU project PSIRP, working on a novel internetworking architecture based on information. He also technical leads a UK-funded project on lifestyle management. Dirk held prior positions as a Chief Researcher with BT Research and as a Principal Scientist at Nokia Research. He holds a Ph.D. degree in Computer Science from Technical University of Aachen, Germany. He published more than 50 peer-reviewed papers in international conferences and journals and has currently 23 international patents.

George Xylomenos is a member of the Mobile Multimedia Laboratory at the Athens University of Economics and Business, Greece, where he is an Assistant Professor of Computer Science. He received his Diploma in Informatics (1993) from the Athens University of Economics and Business, Greece, and M.Sc. (1996) and Ph.D. (1999) degrees in Computer Science from the Department of Computer Science and Engineering at the University of California, San Diego. His research interests include multicast based content distribution, information centric network architectures, provision of Quality of Service over wireless and mobile networks and supporting broadband services over next generation cellular networks.