A Survey of Information-Centric Networking Research

George Xylomenos, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos

***Abstract*—The current Internet architecture was founded upon a host-centric communication model, which was appropriate for coping with the needs of the early Internet users. Internet usage has evolved however, with most users mainly interested in accessing (vast amounts of) information, irrespective of its physical location. This paradigm shift in the usage model of the Internet, along with the pressing needs for, among others, better security and mobility support, has led researchers into considering a radical change to the Internet architecture. In this direction, we have witnessed many research efforts investigating Information-Centric Networking (ICN) as a foundation upon which the Future Internet can be built. Our main aims in this survey are: (a) to identify the core functionalities of ICN architectures, (b) to describe the key ICN proposals in a tutorial manner, highlighting the similarities and differences among them with respect to those core functionalities, and (c) to identify the key weaknesses of ICN proposals and to outline the main unresolved research challenges in this area of networking research.**

***Index Terms*—Information-Centric Networking, Content-Centric Networking, Named-Data Networking, Future Internet, Publish-Subscribe, Internet Architecture**

I. Introduction

THE CURRENT problems of the Internet are a natural consequence of its architecture, which was designed to address the communication needs of a time when a network was needed for sharing rare and expensive resources, such as peripherals, mainframe computers, and long distance communication links. The basic requirement from the Internet at that time was merely that of forwarding packets of data among a limited number of stationary machines, with well-established trust relationships. The key design principles of the Internet made it very simple to link new networks to the Internet and enabled a tremendous growth in its size. In parallel to the Internet's growth, an unprecedented number of innovations, in both the applications and services running on top of it, as well as in the technologies below the (inter-)network layer, have emerged. This is attributed to the hourglass approach followed by the Internet's protocol architecture: the network layer forming the waist of the hourglass is transparent enough, so that almost any application can run on top of it, and simple enough, so that it can run over almost any link-layer technology.

Manuscript received March 28, 2012; revised September 30, 2012 and April 18, 2012.

The authors are with the Mobile Multimedia Laboratory, Department of Informatics, Athens University of Economics and Business, Athens 10434, Greece (e-mail: {xgeorge, chris, vsiris, fotiou, tsilochr, xvas, ntinos, polyzos}@aueb.gr).

Digital Object Identifier 10.1109/SURV.2013.070813.00063

The tremendous growth of the Internet and the introduction of new applications to fulfill emerging needs, has given rise to new requirements from the architecture, such as support for scalable content distribution, mobility, security, trust, and so on. However, the Internet was never designed to address such requirements and in order to help it "evolve" a vicious cycle of functionality patches began appearing, such as Mobile IP. Most of those patches increased the complexity of the overall architecture and proved to be only temporal solutions [1]. In addition, many current and emerging requirements still cannot be addressed adequately by the current Internet. This has raised the question of whether we can continue "patching over patches," or whether a new clean-slate architectural approach for the Internet is actually needed [2]. Along these lines, a research community has been formed which, having identified the limitations of the current Internet, is discussing the key requirements and objectives of the Future Internet, and is proposing new architectures and paradigms to address them.

In this context, *Information-Centric Networking* (ICN) has emerged as a promising candidate for the architecture of the Future Internet. Inspired by the fact that the Internet is increasingly used for information dissemination, rather than for pair-wise communication between end hosts, ICN aims to reflect current and future needs better than the existing Internet architecture. By naming information at the network layer, ICN favors the deployment of in-network caching (or storage, more generally) and multicast mechanisms, thus facilitating the efficient and timely delivery of information to the users. However, there is more to ICN than information distribution, with related research initiatives employing information-awareness as the means for addressing a series of additional limitations in the current Internet architecture, for example, mobility management and security enforcement, so as to fulfill the entire spectrum of Future Internet requirements and objectives.

Although very good survey papers exist for research in the Future Internet area (e.g., [3] and [4]), due to their broad coverage they treat ICN architectures and related research efforts either sketchily or incompletely. The aim of this survey is to focus on ICN and cover the state-of-the-art evenly, broadly, and at some depth. Compared to other ICN surveys (e.g. [5] and [6]) the present survey covers in more detail and depth the most representative and mature ICN architectures and approaches, instead of a subset. In addition to describing the goals and basic concepts of the various research projects on ICN, it identifies the core functionalities of all ICN architectures and highlights their similarities and differences in how

these functionalities are implemented. Furthermore, it provides a critical analysis of the main unresolved research challenges in ICN that require further attention by the community.

The structure of this paper is as follows: In Section II we provide an overview of ICN, presenting the key concepts of ICN architectures and discuss the important problems and limitations of the current Internet that ICN attempts to solve. In Section III we identify the core functionalities of all ICN architectures and then present the most mature and representative ICN approaches. In Section IV we discuss the similarities and differences among ICN architectures, with respect to the implementation of these core functionalities. In Section V, we outline the main challenges that remain unresolved for researchers interested in this area of networking research. Finally, a brief summary and our conclusions are provided in Section VI.

II. Information-Centric Networking: Terminology, Concepts and Overview

In this section we introduce the key concepts and principles of ICN and discuss how each one of them aims to address some of the current Internet's problems and limitations. This discussion also sets the framework for examining in detail each proposed ICN architecture in Section III.

A. Focus on Information Naming

The Internet has been transformed from an academic network to a global infrastructure for the massive distribution of information, with over 1 billion of connected devices [7], 1 trillion of indexed web pages [8] and Exabytes of annually transferred data [7]. Users are more and more interested in receiving information/content/data[1] wherever it may be located, rather than in accessing a particular computer system (host or server). However, the fact that the Internet is still based on an underlying host-centric communication model requires the user to specify in each request not only the desired information, but also the specific server from which it can be retrieved from. Unless add-on functionality is used, the Internet's native network-layer mechanisms cannot locate and fetch the requested information from the optimal location where it is hosted, unless the user somehow knows and includes the optimal location in the request. In the research community, the first ideas for shifting from the host-centric network design to an information-centric network design were introduced almost a decade ago in the seminal papers of Gritter and Cheriton [9] (in the context of the TRIAD project [10]) and Carzaniga et al. [11], [12], [13]. Other works of that period also realized the need for information-centric communication, assuming however that it would operate as an overlay on top of the current architecture, based on the functionality offered by *Distributed Hash Tables* (DHTs) [14], [15].[2]

The ICN approach fundamentally decouples information from its sources, by means of a clear location-identity split. The basic assumption behind this is that information is named, addressed, and matched independently of its location, therefore it may be located anywhere in the network [18], [19]. In ICN, instead of specifying a source-destination host pair for communication, a piece of information itself is named. An indirect implication (and benefit) of moving from the host naming model to the information naming model, is that information retrieval becomes receiver-driven. In contrast to the current Internet where senders have absolute control over the data exchanged, in ICN no data can be received unless it is *explicitly* requested by the receiver. In ICN, after a request is sent, the network is responsible for locating the best source that can provide the desired information. Routing of information requests thus seeks to find the best source for the information, based on a location-independent name.

B. Focus on Information Delivery

The shift towards content-centric bandwidth-demanding applications requires the Internet to efficiently deliver massive amounts of information and handle large spikes or surges in traffic, commonly referred to as *flash crowds*. However, the data-agnostic Internet architecture lacks native mechanisms for handling flash crowd events and for enabling efficient information delivery. In the current Internet, data in transit are treated by network elements as a series of bytes that have to be transferred from a specific source to a specific destination and, as such, network elements have no knowledge of the information they transfer and hence cannot realize optimizations that would otherwise be possible (e.g., smart in-network caching, information replication at various points, information-aware traffic engineering). For the Internet to fully exploit the existing in-network storage capabilities, it must be extended with in-network information-aware mechanisms for the identification and retrieval of information from its optimal location [20]. The emergence of such techniques at the application level (e.g., web caching) only confirms that they were actually an afterthought for the Internet.

Content Delivery Networks (CDNs) deployed as overlays, apply these techniques over the wide area at the application layer but, especially for the case of flash crowds, the amount, location, and destination of traffic cannot always be anticipated and the investment for having a CDN to accommodate all possible cases is certainly not viable, especially given the constant rise in user-generated information. Moreover, CDNs typically employ network-unaware mechanisms, which lead to inefficient utilization of the underlying network resources. Instead, an Internet-wide infrastructure supporting in-network mechanisms for efficient information retrieval would be preferable. Recent work [21] based on empirical evidence and data analysis from Open CDNs shows that some flash crowds can grow too quickly for application layer dynamic resource allocation to keep up, e.g., by the CDN reallocating cache resources, and that the problem becomes worse when the flash crowd is related to uncacheable (dynamic) information.

In ICN the network may satisfy an information request not only through locating the original information source, but also by utilizing (possibly multiple) in-network caches that hold copies of the desired information (or pieces of

[1]The terms *information*, *content* and *data* will be used interchangeably and with the same meaning in the rest of the manuscript, as in [6].

[2]Subsequent works also explored the direct application of DHTs to information-centric communication, without an underlying routing protocol [16], [17].

it). This can be accomplished without resorting to add-on, proprietary and costly overlay solutions (e.g., CDNs), since the network layer in ICN operates directly on named information. ICN-based architectures see non-opaque data packets, in the sense that these are named based on the information they carry. Therefore, information fragments (packets in current terms) can be cached and retrieved easily, unlike in the current Internet where costly techniques like *Deep Packet Inspection* (DPI) would have to be used for answering requests with data/packets cached on the routers [22], [23], [24], not to mention that DPI does not work when packets are encrypted. Moreover, by naming information, ICN allows the aggregation of requests for the same information, thus facilitating its delivery to the corresponding destinations via multicast forwarding. Finally, access controls (i.e., *who* is allowed to access *which* data) can potentially be applied directly at the network layer, since network elements are aware of what information is being transferred inside each packet.

C. Focus on Mobility

The addressing scheme of the Internet was designed with fixed hosts in mind, since a host's IP address must belong to the network where the host is currently attached. However, statistics show a constantly increasing number of non-fixed hosts accessing the Internet, with forecasts saying that by 2015, traffic from wireless terminals will exceed traffic from wired ones [7]. Wireless and mobile devices may easily switch networks, changing their IP address and thus introducing new communication modes based on intermittent and, possibly, opportunistic connectivity. However, such an approach does not achieve continuous connectivity while on the move, which is becoming an increasingly important requirement.

On the other hand, the Mobile IP protocol, a patch to remedy the problem of locating moving hosts, imposes "triangular routing": packets first need to be routed to a home agent, representing the mobile host at its home network, and from there to the current location of the mobile node via a tunnel. This is a major inefficiency, since traffic has to travel along a path longer than the optimal, a problem significantly aggravated when the mobile node, its home agent, and the third party that the host is communicating with are all located in distant *Autonomous Systems* (AS). Even traffic originating from a mobile node may need to be tunneled via its home agent, since many routers on the Internet exercise ingress filtering, i.e., they check that incoming traffic comes from the actual network it claims to originate from, meaning that the mobile node may not be able to directly send traffic from its current location using its permanent home address. Mobile IP, just like overlay networks [25], also tends to violate the usual "valley free" *Border Gateway Protocol* (BGP) routing policies, since packets are first routed to the mobile node's home agent and from there re-routed to its currently hosting network. This leads to (a) "valley routing", i.e., a client AS (where the home agent is located) serves traffic for a provider AS, and (b) "exit policy violation", i.e., traffic exiting from an exit point different than the one it was supposed to, according to the BGP rules for a given traffic destination.

In ICN, host mobility is addressed by employing the publish/subscribe communication model [26]. In this model, users interested in information *subscribe* to it, i.e., they denote their interest for it to the network, and users offering information *publish* advertisements for information to the network. Inside the network, *brokers* are responsible for matching subscriptions with publications i.e., they provide a *rendezvous* function. It is important to note that the publish/subscribe terminology used in the context of ICN (e.g., [27]) differs from that of traditional publish/subscribe systems (e.g., [11], [12], [13], [26]). In traditional publish/subscribe systems, *publish* involves the actual transmission of data while *subscribe* results in receiving data published in the future, with the ability of receiving previously published data being optional. In ICN, on the other hand, publish involves only announcement of the availability of information to the network, whereas subscriptions by default refer to already available information, leaving the option of permanent subscriptions (i.e., receiving multiple publications matching a single subscription) as optional.

The strength of the publish/subscribe communication model stems from the fact that publication and subscription operations are decoupled in time and space [28]. The communication between a publisher and a subscriber does not need to be time-synchronized, i.e., the publisher may publish information before any subscribers have requested it and the subscribers may initiate information requests after publication announcements. Publishers do not usually hold references to the subscribers, neither do they know how many subscribers are receiving a particular publication and, similarly, subscribers do not usually hold references to the publishers, neither do they know how many publishers are providing the information [26]. These properties allow for the efficient support of mobility: mobile nodes can simply reissue subscriptions for information after handoffs and the network may direct these subscriptions to nearby caches rather than the original publisher.

D. Focus on Security

The Internet was designed to operate in a completely trustworthy environment. User and data authentication, data integrity and user privacy were not a requirement; indeed the focus was on openness and flexibility in allowing new hosts to join the network. Moreover, the Internet was designed to forward any traffic injected in the network, resulting in an imbalance of power between senders and receivers. These characteristics allow spammers, hackers and attackers in general to launch *Denial of Service* (DoS) attacks against the Internet infrastructure or against Internet hosts and services, while easily covering their tracks. In order to cope with such malicious and/or selfish behavior, add-on security patches and trust mechanisms have been developed, such as firewalls and spam filters, as well as new security protocols that complement the existing (inter)networking protocols (e.g., IPSec and DNSSec). However, such solutions do not penetrate deep into the network and bad data still gets forwarded, clogging systems and possibly fooling filtering mechanisms [3]. The required processing overhead and the Internet's end-to-end philosophy have so far prevented placing security and trust mechanisms deeper into the network, where it would be most effective in avoiding or identifying and stopping attacks.

Many of the security problems of the Internet are largely due to the disconnection between information semantics at the

application layer and the opaque data in individual IP packets. This places a significant burden on integrating accountability mechanisms into the overall architecture. Point solutions like DPI or lawful interception try to restore this broken link between the actual information semantics and the data scattered in individual packets. However, this is achieved at a relatively high cost and is therefore only applicable to critical problems, such as law enforcement. As a result, while secure end-to-end connections are prevalent, the overall Internet architecture is still not self-protected against malicious attacks and data is not secure. At the same time, the lack of an accountability framework which would allow non-intrusive and non-discriminatory means to detect misbehavior and mitigate its effects, while retaining the broad accessibility to the Internet and ensuring both data security and communication privacy (i.e., hiding from non-authorized parties that a communication between two parties took place) is a crucial limitation to overcome [20].

ICN architectures are in contrast interest-driven, i.e., there is no data flow unless a user has explicitly asked for a particular piece of information. This is expected to significantly reduce the amount of unwanted data transfers (such as spam) and also facilitate the deployment of accountability and forensic mechanisms on the network points that handle "availability" and "interest" signaling. Moreover, for ICN architectures that use self-certifying names for information, malicious data filtering will be possible even by in-network mechanisms. Finally, most ICN architectures add a point of indirection between users requesting a piece of information and users possessing this piece of information, decoupling the communication between these parties. This decoupling can be a step towards fighting denial of service attacks, as requests can be evaluated at the indirection point, prior to arriving to their final destination. Indirection can also benefit user privacy, as a publisher does not need to be aware of the identities of its subscribers.

III. ICN Approaches

The various existing ICN initiatives focus on designing an Internet architecture that will replace the current host-centric model and will directly address the problems and limitations identified in the previous section. ICN oriented projects (see Figure 1) include the DONA [29] project at Berkeley, the EU funded projects Publish-Subscribe Internet Technology (PURSUIT) [30] and its predecessor Publish-Subscribe Internet Routing Paradigm (PSIRP) [31], Scalable & Adaptive Internet soLutions (SAIL) [32] and its predecessor 4WARD [33], COntent Mediator architecture for content-aware nETworks (COMET) [34], CONVERGENCE [35], the US funded projects Named Data Networking (NDN) [36] and its predecessor Content Centric Networking (CCN) [37] and MobilityFirst [38], as well as the French funded project ANR Connect [39] which adopts the NDN architecture.

Although they are still under active development, these ICN architectures address a set of key functionalities, albeit with different approaches. Below we identify these key functionalities, which will form the basis for presenting and comparing the various ICN initiatives in the remainder of the paper.

- **Naming**: The structure of the name assigned to a piece of information (or service) that can be communicated over the network is one of the main characteristics of each ICN architectural proposal. In all ICN architectures information names are location-independent. On the other hand, depending on the approach, names may range from flat to hierarchical and may or may not be human-readable.
- **Name resolution and data routing**: Name resolution involves matching an information name to a provider or source that can supply that information, while data routing involves constructing a path for transferring the information from that provider to the requesting host. A key issue is whether these two functions are integrated, or *coupled*, or are independent, or *decoupled*. In the coupled approach, the information request is routed to an information provider, which subsequently sends the information to the requesting host by following the reverse path over which the request was forwarded. In the decoupled approach, the name resolution function does not determine or restrict the path that the data will use from the provider to the subscriber. For example, an independent data routing module may send to the provider a source route to the requesting host.
- **Caching**: We distinguish between *on-path* and *off-path* caching. In on-path caching the network exploits information cached along the path taken by a name resolution request, while in off-path caching the network exploits information cached outside that path. In ICN architectures with decoupled name resolution and data routing, off-path caching must be supported by the name resolution system, which handles caches as regular information publishers. If name resolution and data transfer are coupled, off-path caching must be supported by the routing system used to forward the requests for information.
- **Mobility**: Subscriber mobility is intrinsically supported in ICN architectures, since mobile subscribers can just send new subscriptions for information after a handoff. Publisher mobility is more difficult to support, since the name resolution system (in the coupled approach) or the routing tables (in the decoupled approach) need to be updated.
- **Security**: This aspect is tightly related to the naming structure [40]. On the one hand, human-readable names require a trusted agent or a trust relationship with the name resolution system to verify that the returned information corresponds to the requested name. On the other hand, flat names can support self-certification, but are not-human readable, thus requiring another trusted system to map human-readable names to flat names.

It is important to note that these are not ICN-specific functionalities, but rather the common core of all the ICN architectures considered. As such, this list simply aims to assist in shaping the presentation of each individual ICN architecture in the remainder of this section, as well as the ensuing discussions in the following sections.

A. DONA

While many projects, starting with TRIAD [10], proposed extending the Internet with content routing capabil-

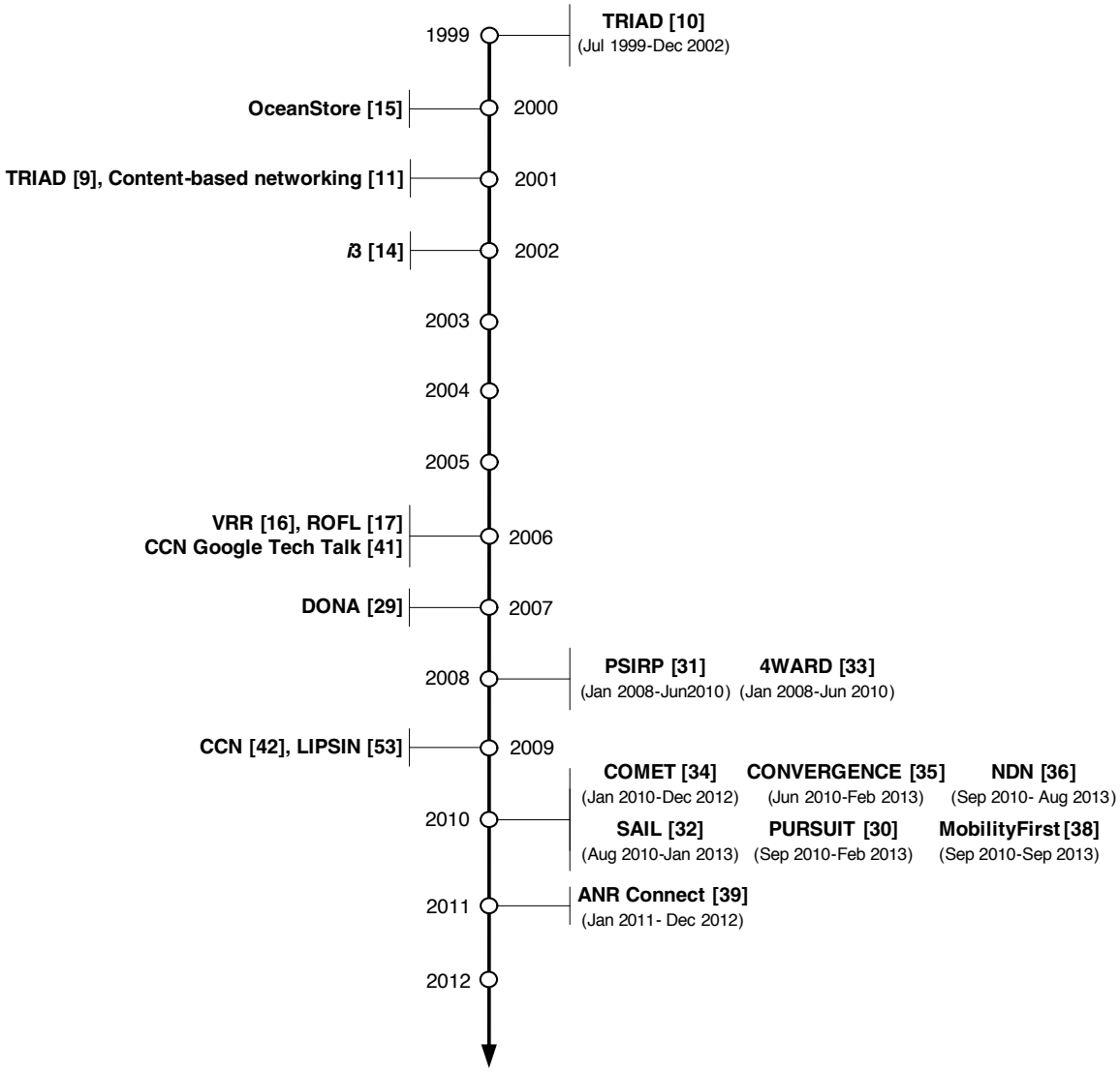

Fig. 1. Timeline of key ICN milestones. Seminal ICN papers are shown on the left hand side, while ICN-related projects are shown on the right hand side.

ities (see Figure 1), the *Data Oriented Network Architecture* (DONA) [29] from UC Berkeley is one of the first complete ICN architectures, as it radically changes naming by replacing the hierarchical URLs with flat names. Unlike URLs which are bound to specific locations via their DNS component, the flat names in DONA can be persistent, even if the information moves. This allows information to be cached and replicated at the network layer, thus increasing information availability. Finally, names in DONA allow users to verify that the received information matches a requested name via cryptographic techniques. On the other hand, DONA maintains IP addressing and routing, either globally or locally (see below), deploying a name resolution mechanism as an overlay that maps its flat names to the corresponding information.

1) Naming: In DONA each piece of information (or service) is associated with a *principal*. Names consist of the cryptographic hash of the principal's public key `P` and a label `L` uniquely identifying the information with respect to the principal. Naming granularity is left to the principals, who are considered to be the owners of the corresponding information. For instance, principals may name either an entire web site or each individual web page within it. Names are flat, application-independent, location-independent and globally unique. For immutable data the label can be the cryptographic hash of the information object itself, thus allowing any purveyor (e.g., a CDN) to offer such data. Clients interested in an information object are assumed to learn its name through some trusted external mechanisms (e.g., a search engine). Unlike structured DNS names, flat names in DONA do not embed a fixed administrative structure, thus they are easy to map to any private namespace of human-readable names.

2) Name Resolution and Data Routing: Name resolution in DONA is provided by specialized servers called *Resolution Handlers* (RHs). There is at least one logical RH at each AS. RHs are interconnected, forming a hierarchical name resolution service on top of the existing inter-domain routing relations, as shown in Figure 2, so as to allow name resolution and data routing to respect the established routing policies between AS's. In order to make an information object available, the publisher (principal) sends a REGISTER message with the object's name to its local RH, who stores a pointer to the

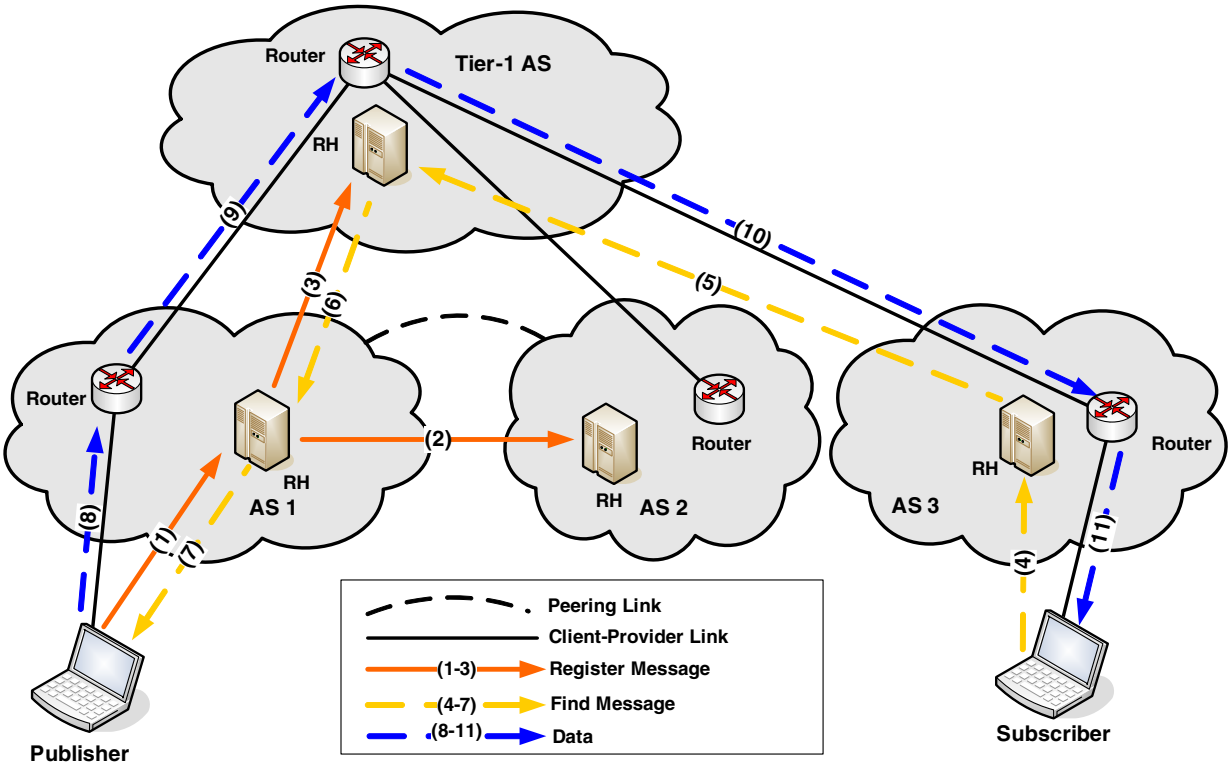

Fig. 2. The DONA architecture. RH stands for Resolution Handler.

principal (arrow 1). The RH then propagates this registration to the RHs in its parent and peering domains, following the established routing policies (arrows 2-3), causing each intermediate RH to store a mapping between the object's name and the address of the RH that forwarded the registration. As a result, registrations are replicated in RHs all the way up to the tier-1 providers and, since all tier-1 providers are peers with each other, RHs located at tier-1 providers are aware of *all* registrations in the entire network. Publishers can also issue wildcard REGISTER messages to notify the RH hierarchy that they can provide all possible data items for a specific principal.

In order to locate an item, a subscriber sends a FIND message to its local RH, which also propagates this message to its parent according to its routing policies, until a matching registration entry is found (arrows 4-5). At that point, requests follow the pointers created by the registrations in order to eventually reach the publisher (arrows 6-7). Since tier-1 providers are aware of all objects in the network, this process is guaranteed to succeed if the requested name exists. Subscribers can also issue wildcard FIND messages to ask for immutable data with a specific label, regardless of its purveyor.

Data routing can be either decoupled or coupled with name resolution. In the decoupled option, when a FIND message reaches an appropriate publisher, the data can be directly sent to the subscriber using regular IP routing and forwarding. The actual data transmission will follow the established routing policies for traffic between the publisher's and the subscriber's AS's. This requires global IP addresses, which are nearly exhausted, so DONA also offers a coupled option which relies on domain-local IP addresses only. In the coupled option the FIND messages gather path-labels as they move from RH to RH, indicating the sequence of AS's crossed by the request. When the request reaches the publisher, these path-labels are simply used in the reverse order to retrace the path towards the subscriber (arrows 8-11). While the coupled option obviates the need for global IP addresses and large BGP routing tables, since all path-labels have a local meaning, it enforces symmetric routing (at least, at the AS level) between requests and responses, which is not necessarily the case with regular BGP routing.

DONA can also support multicast channels, by allowing FIND messages to be cached in RHs for a specified period of time and sending information updates in response to these messages until they expire. When additional FIND messages for the same information are received by the RH, they are merged into a single entry with multiple path-labels for the responses, thus creating a multicast distribution tree. Note that this can only work with the coupled option, as it requires data to follow the reverse AS path taken by the FIND messages.

3) Caching: DONA supports on-path caching via the RH infrastructure. A RH that decides to cache a requested data object can replace the source IP address of an incoming FIND request with its own IP address, before forwarding the message to the next RH. As a result, any response will surely traverse the current RH, thus the data returned will be cached there. If path-labels are used, the data always return via the intermediate RHs, who can then decide whether to cache the information or not. If a subsequent FIND message requesting the same object reaches a caching RH, the RH can directly return the data to the subscriber. Information may also be replicated off-path, provided that each purveyor registers the information through its local RH. A RH receiving multiple

REGISTER messages for the same information maintains (and propagates upwards) only the pointers to the best available copy (e.g., the closest one).

4) Mobility: Mobile subscribers can simply issue new FIND messages from their current location, relying on the RH infrastructure to provide them with the closest copy of the information. Mobile publishers can also unregister and re-register their information when changing their network location, but this incurs a non-negligible messaging overhead, since these messages need to be relayed all the way to the tier-1 RHs to ensure that (a) no stale registration state will exist and (b) that the advertised information is traceable to its new location.

5) Security: Names in DONA are self-certifying, i.e., they allow the subscriber to verify that the data received matches the name requested. For mutable data, a client requesting an information object named `P:L` will also receive as metadata the public key of the principal (which is bound to `P` via its hash) and a signature for the data object itself, thus allowing the data to be authenticated as coming from the specific principal. For immutable data, the subscriber can simply verify that the label `L` is indeed the cryptographic hash of the information object, regardless of the purveyor acting as the principal. This allows subscribers to choose a purveyor according to its reputation and performance.

The design of DONA can either prevent or mitigate a series of attacks to the RH infrastructure. A RH will only accept information registrations by authenticated principals (bound to `P`) or purveyors of immutable information (bound to `L`) and it will only accept forwarded registrations from trusted RHs, since all traffic follows established routing policies. Providers can enforce contractual limits on REGISTER and FIND messages from customers to guard against control-plane resource exhaustion attacks. In order to protect customer AS's from misbehaving RHs, DONA allows explicit requests for access to copies other than the closest one. Finally, DONA delegates the responsibility for guarding against user-plane resource exhaustion attacks to existing IP mechanisms.

B. NDN

The *Content Centric Networking* (CCN) [37] architecture from PARC is the other pioneering fully-fledged ICN architecture. Its basic ideas were described in a Google tech talk [41], long before the first paper describing the CCN architecture was published [42] (see Figure 1). The *Named Data Networking* (NDN) [36] project, funded by the US Future Internet Architecture program, is further developing the CCN architecture. NDN envisions reshaping the Internet protocol stack by making the exchange of *named data* the thin waist of the Internet architecture, using various networking technologies below the waist for connectivity, including, but not limited to, IP. In NDN a *strategy* layer mediates between the named data layer and the underlying network technologies to optimize resource usage, e.g., to select a link in a multi-homed node, while a *security* layer applies security functionalities directly on named data. A crucial aspect of NDN is that names are hierarchical, thus allowing name resolution and data routing information to be aggregated across similar names, something considered to be critical for the scalability of the architecture.

1) Naming: Names in NDN are hierarchical and *may* be similar to URLs, for example, an NDN name can be `/aueb.gr/ai/main.html`. However, NDN names are *not* necessarily URLs: their first part is not a DNS name or an IP address and they do not have to be human-readable. Instead, in NDN each name component can be anything, including a dotted human-readable string or a hash value. In NDN a request for a name is considered to match any piece of information whose name has the requested name as a prefix, for example, `/aueb.gr/ai/main.html` can be matched by an information object named `/aueb.gr/ai/main.html/_v1/_s1`, which could mean the first segment of the first version of the requested data. After receiving this information object, the subscriber could ask for the next data segment either directly by requesting `/aueb.gr/ai/main.html/_v1/_s2`, or for the next sibling under this version. Alternatively, the subscriber could ask for the next version by requesting the first sibling of `/aueb.gr/ai/main.html/_v1`. While the way information objects are segmented is expected to be known by the subscriber's application, the prefix matching rule enables an application to discover what is available. Furthermore, it allows the subscriber to ask for data that have not been produced yet: a publisher can advertise that it can satisfy requests for a specific prefix, and then return information objects with complete NDN names. This can be used to implement various applications where information objects are generated dynamically, hence their full names cannot be known in advance, such as voice conferencing [43].

2) Name Resolution and Data Routing: In NDN subscribers issue INTEREST messages to request information objects which arrive in the form of DATA messages, with both types of message carrying the name of the requested/transferred information object. As shown in Figure 3, all messages are forwarded hop-by-hop by *Content Routers* (CRs), with each CR maintaining three data structures: the *Forwarding Information Base* (FIB), the *Pending Interest Table* (PIT) and the *Content Store* (CS). The FIB maps information names to the output interface(s) that should be used to forward INTEREST messages towards appropriate data sources. The PIT tracks the incoming interface(s) from which pending INTEREST messages have arrived, i.e., those INTEREST messages for which matching DATA messages are expected. Finally, the CS serves as a local cache for information objects that have passed through the CR.

When an INTEREST arrives, the CR extracts the information name and looks for an information object in its CS whose name matches the requested prefix. If something is found, it is immediately sent back through the incoming interface in a DATA message and the INTEREST is discarded. Otherwise the router performs a longest prefix match on its FIB in order to decide towards which direction this INTEREST should be forwarded. If an entry is found in the FIB, the router records the INTEREST's incoming interface in the PIT and pushes the INTEREST to the CR indicated by the FIB. In Figure 3, the subscriber sends an INTEREST for the name `/aueb.gr/ai/new.htm` (arrows 1-3). If the PIT already contains an entry for the *exact* name, meaning that this *exact*

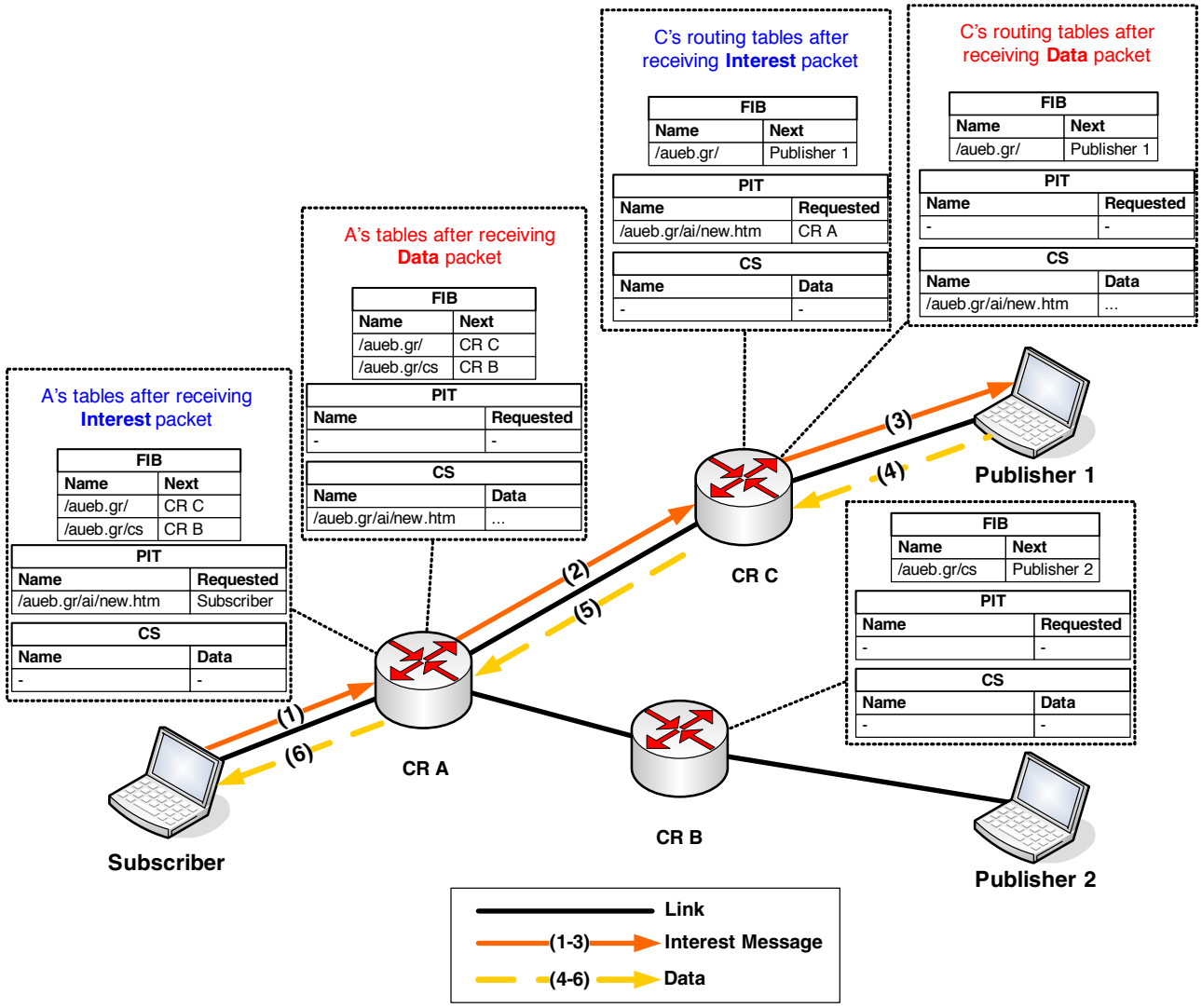

Fig. 3. The NDN architecture. CR stands for Content Router, FIB for Forwarding Information Base, PIT for Pending Interest Table, CS for Content Store.

information object had already been requested, the router adds the incoming interface to this PIT entry and discards the INTEREST, effectively forming a multicast tree for the information object.

When an information object that matches the requested name is found at a publisher node or a CS, the INTEREST message is discarded and the information is returned in a DATA message. This message is forwarded back to subscriber(s) in a hop-by-hop manner, based on the state maintained in the PITs. Specifically, when a CR receives a DATA message, it first stores the corresponding information object in its CS and then it performs a longest-prefix match in its PIT to locate an entry matching the DATA packet;[3] if a PIT entry lists multiple interfaces, the DATA message is duplicated, thus achieving multicast delivery. Finally, the CR forwards the DATA message packet to these interfaces and deletes the entry from the PIT (arrows 4-6). In case there are no matching entries in the PIT, the router discards the DATA packet as a duplicate.

In NDN name resolution and data routing are coupled, since DATA messages follow the pointers left in the PITs by INTEREST messages, therefore routing is by definition symmetric. In order to populate the FIBs, NDN can use distributed routing protocols like OSPF [42], in which CRs advertise name prefixes rather than IP address ranges, e.g., a router could advertise `/aueb.gr` to inform the network that it can provide information objects whose prefix is `/aueb.gr`. A CR may have multiple interfaces in its FIB for a prefix, for example, if it is multi-homed or if it is aware of multiple CDN servers hosting the information. In this case its strategy layer may choose to send the INTEREST either to all these interfaces (if multiple DATA messages are returned, all but the first are automatically discarded) or only to the interface that has exhibited the best performance so far.

3) Caching: NDN natively supports on-path caching, since each CR first consults its CS whenever it receives an INTEREST message and caches all information objects carried by DATA messages. The CS can use LRU or any other replacement policy, but, realistically, it cannot be used for long-term storage if it just caches everything it sees [44], [45], therefore it is mostly useful for recovery from packet losses

[3]The longest-prefix match is needed since the requested name may be a prefix of the one returned.

and for handling flash crowds, where many users request the same data in close succession. Off-path caching is supported by delivering an INTEREST to any data source that may be hosting the requested information object, e.g., the strategy layer can direct the INTEREST to a CDN server rather than to the originating publisher. This is not transparent to NDN however, as it requires populating the FIBs with pointers to such copies, which in turn requires the name prefixes of these copies to be advertised by the CDN server through the routing protocol used.

4) Mobility: When a subscriber moves in NDN, it can simply issue new INTEREST messages from its current location for the information objects it has not yet received. These will be suppressed by the PIT of the first common CR in both delivery routes (prior and post the handoff). Of course, the corresponding information objects will also be delivered to its old location. When a publisher moves on the other hand, the FIBs pointing to it have to be updated, which requires advertising again the name prefixes for the information it is hosting via the routing protocol. As this represents a very high overhead in high-mobility solutions, NDN utilizes the *Listen First Broadcast Later* (LFBL) protocol [46] to implement mobility in ad-hoc/opportunistic networks. In LFBL, INTEREST messages are flooded. When a potential source for the requested information receives an INTEREST, it listens to the (wireless) channel in order to discover if another node has already sent a matching DATA message. If not, it sends the DATA message itself towards the subscriber.

5) Security: NDN supports the association of human-readable hierarchical information names with the corresponding information objects in a verifiable way [47]. Each DATA message contains a signature over the name and the information included in the message, plus information about the key used to produce the signature, e.g., the public key of the signer, a certificate for that public key or a pointer to them. This allows any node, including CRs, to verify the binding between the (possibly, human-readable) name of the packet and the accompanying information. In order to verify that the information comes from an authorized source though, the subscriber must trust the owner of the public key used for signing. The hierarchical structure of names simplifies building trust relationships, for example, `/aueb.gr/ai/main.html` may be signed by the owner of the `/aueb.gr/ai` domain, whose key may be certified by the owner of the `/aueb.gr` domain. NDN also supports anonymous operation by using a Tor-like approach named ANDaNA [48].

C. PURSUIT

The *Publish Subscribe Internet Routing Paradigm* (PSIRP) [31] project and its continuation the *Publish Subscribe Internet Technology* (PURSUIT) [30] project (see Figure 1), both funded by the EU Framework 7 Programme, have produced an architecture that completely replaces the IP protocol stack with a publish-subscribe protocol stack. The PURSUIT architecture consists of three separate functions: *rendezvous*, *topology management* and *forwarding*. When the rendezvous function matches a subscription to a publication, it directs the topology management function to create a route between the publisher and the subscriber. This route is finally used by the forwarding function to perform the actual transfer of data.

1) Naming: Information objects in PURSUIT are identified by a (statistically) unique pair of IDs, the *scope ID* and the *rendezvous ID*. The scope ID groups related information objects while the rendezvous ID is the actual identity for a particular piece of information [49]. Information objects may belong to multiple scopes (possibly with different rendezvous IDs), but they must always belong to at least one scope. Scopes serve as a means of (a) defining sets of information objects within a given context and (b) enforcing "boundaries" based on some dissemination strategy for the scope. For example, a publisher may place a photograph under a "friends" scope and a "family" scope, with each scope having different access rights. While PURSUIT names are flat as in DONA, scopes in PURSUIT can be organized in scope graphs of variable forms, including hierarchies, therefore a complete name consists of a sequence of scope IDs and a single rendezvous ID, thus generalizing the DONA naming scheme.

2) Name Resolution and Data Routing: Name resolution in PURSUIT is handled by the rendezvous function, which is implemented by a collection of *Rendezvous Nodes* (RNs), the *Rendezvous Network* (RENE), implemented as a hierarchical DHT [50], [51], as shown in Figure 4. When a publisher wants to advertise an information object, it issues a PUBLISH message to its local RN which is routed by the DHT to the RN assigned with the corresponding scope ID (arrows 1-2).[4] When a subscriber issues a SUBSCRIBE message for the same information object to its local RN, it is routed by the DHT to the same RN (arrows 3-6). The RN then instructs a *Topology Manager* (TM) node to create a route connecting the publisher with the subscriber for data delivery (arrows 7-8). The TM sends that route to the publisher in a START PUBLISH message (arrows 9-10), which finally uses this route to send the information object via a set of *Forwarding Nodes* (FNs).

The TM nodes in PURSUIT jointly implement the topology management function by executing a distributed routing protocol to discover the network topology, e.g., OSPF. The actual delivery paths are calculated upon request by the rendezvous function as a series of links between FNs and encoded into source routes using a technique based on Bloom filters [52]. Specifically, each network node assigns a tag, i.e., a long bit string produced by a set of hash functions, to each of its outgoing links, and advertises these tags via the routing protocol. A path through the network is then encoded by ORing the tags of its constituent links and the resulting Bloom filter is included in each data packet. When a data packet arrives at a FN, the FN simply ANDs the tags of its outgoing links with the Bloom filter in the packet; if any tag matches, then the packet is forwarded over the corresponding link [53]. In this manner, the only state maintained at the FNs is the link tags. Multicast transmission can be achieved by simply encoding the entire multicast tree into a single Bloom filter.

Subsequent packets belonging to the same information

[4]Note that the RN assigned with a scope ID may reside outside the AS of its publisher, due to the way DHTs operate.

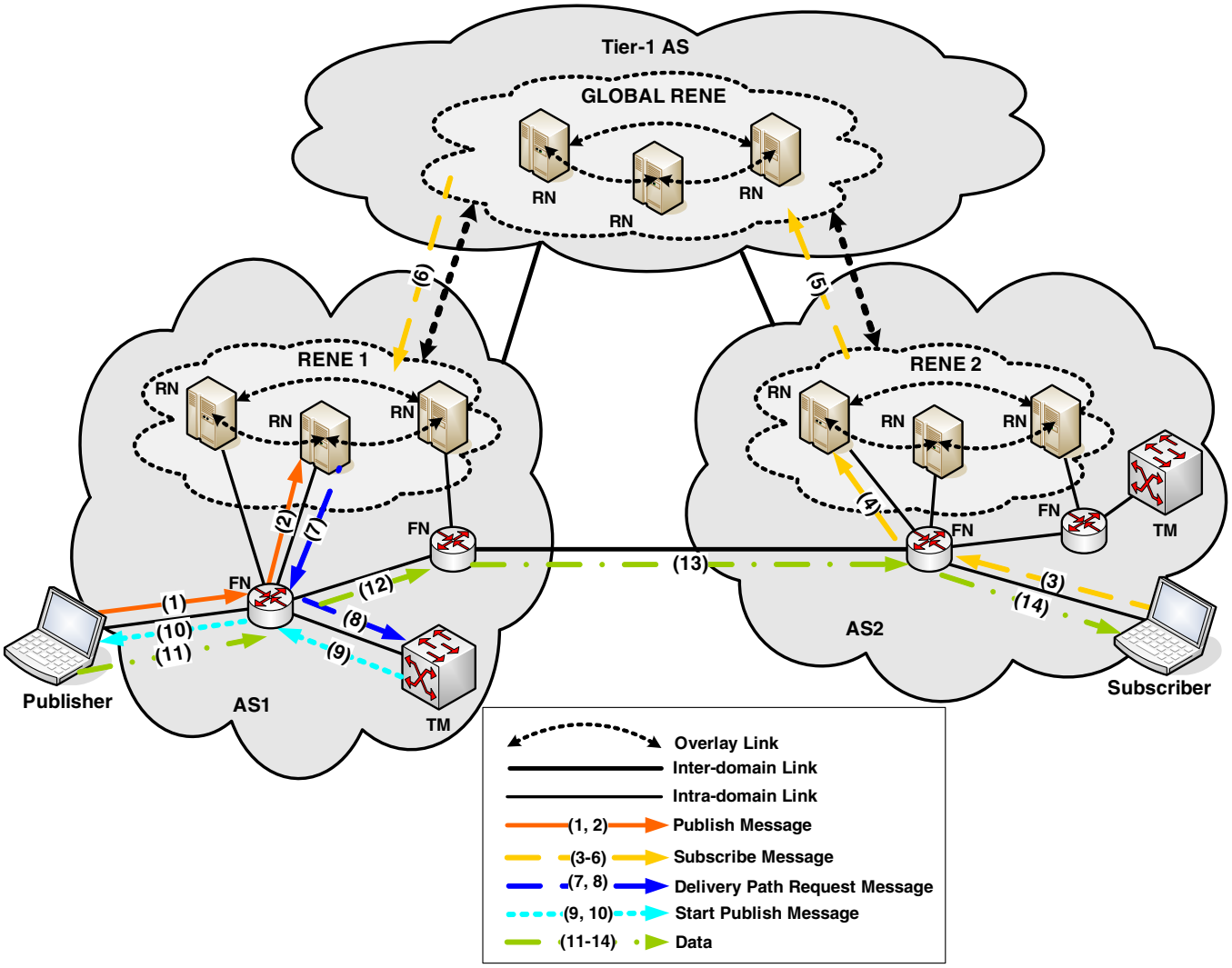

Fig. 4. The PURSUIT architecture. RN stands for Rendezvous Node, RENE for REndezvous NEtwork, FN for Forwarding Node and TM for Topology Manager.

object can be individually requested by the subscriber using the notion of *Algorithmic IDs*, i.e., packet names generated by an algorithm agreed by the communicating entities. These requests are forwarded similarly to data packets, using reverse Bloom filters calculated by the TM to bypass the RENE. This allows the realization of transport layer protocols, e.g., via a sliding window of pending requests.

Name resolution and data routing are decoupled in PURSUIT, since name resolution is performed by the RENE, while data routing is organized by the TMs and executed by the FNs. While name resolution can be time consuming, especially since DHT routing does not follow the shortest paths between the communicating nodes, data forwarding can take place at line speeds, without placing any state at the FNs [53]. Furthermore, the separation of routing and forwarding allows the TMs to calculate paths using complex criteria (e.g., load balancing), without requiring signaling to the (stateless) FNs. On the other hand, the topology management and forwarding functions as described are only adequate for the intra-domain case and need to be extended (e.g., with label switching) for the inter-domain level.

3) Caching: PURSUIT can support both on-path and off-path caching [54]. In the on-path case, forwarded packets are cached at FNs in order to potentially serve subsequent requests. However, on-path caching may not be very effective due to the decoupled nature of name resolution and data routing, since requests for the same information object can reach the same RN, whereas the actual data transfers may use entirely different paths. In the off-path case, caches operate as publishers, by advertising the available information to the RENE. Managed information replication, as in CDNs, can also be efficiently supported by the PURSUIT architecture [55].

4) Mobility: Mobility in PURSUIT is greatly facilitated by the use of multicast and caching [54]. Four types of mobility cases are considered, based on host movement (local, global) and technology interchange (static when a single technology is involved, dynamic when vertical handoffs are performed). Local subscriber mobility can be handled via multicast and caching, i.e., by multicasting information objects to multiple possible locations for the mobile subscriber and receiving information objects from nearby caches after a handoff. Global subscriber mobility is handled by modifying the forwarding

function of the architecture [56]. Mobility prediction can be used to reduce handoff latencies by caching information requested by the subscriber to the areas where the subscriber is expected to move after a handoff [54]. Publisher mobility is harder, since the topology management function has to be notified of the publisher's new position in the network.

5) Security: PURSUIT supports the *Packet Level Authentication* (PLA) technique [57] for encrypting and signing individual packets. This technique assures data integrity and confidentiality as well as malicious publisher accountability. PLA can be used to check packets either at FNs or at their final destination. The use of flat names also permits self-certifying names for immutable data objects, using the object's hash as the rendezvous ID. Moreover, paths encoded into Bloom filters can use dynamic link identifiers, making it impossible for an attacker to craft Bloom filters or even to reuse old Bloom filters to launch DoS attacks. Finally, security solutions for mitigating spam [58] and for preserving privacy have been developed for PURSUIT.

D. SAIL

The *Architecture and design for the future Internet* (4WARD) [33] project and its continuation *Scalable and Adaptive Internet Solutions* (SAIL) [32] (see Figure 1), both funded by the EU Framework 7 Programme, are investigating designs for the Future Internet and ways to facilitate a smooth transition from the current Internet. While both projects have a very wide scope, in this survey we will focus on one of their research areas, the *Network of Information* (NetInf), which designs an ICN architecture that supports the exchange of named information objects. Beyond the aspects covered below, the SAIL architecture[5] includes many other services, such as searching for information objects via keywords. The SAIL architecture is very general: it combines elements present in the NDN and PURSUIT approaches and can even operate in a hybrid mode. Furthermore, it can be implemented over different routing and forwarding technologies, by introducing *convergence layers* to translate SAIL messages to actual network packets [59].

1) Naming: Information object names in SAIL are "flat-ish": they provide some structure and they can even be hierarchical, but they do not carry location or organizational information. SAIL defines the `ni://A/L` URI scheme in which names consist of an authority part `A` and a local (with respect to the authority) part `L`. Each part can be a hash, thus allowing for self-certification, or any other type of string, thus allowing for regular URLs [60]. SAIL names are considered flat for name comparison purposes, that is, a subscription will only match a publication if there is an exact name match between them, as in PURSUIT. On the other hand, SAIL names can be considered hierarchical when used for routing, that is, routers can use longest prefix matching to determine how to route a message, as in NDN [59].

2) Name Resolution and Data Routing: Name resolution and data routing can be either coupled or decoupled in SAIL, and even hybrid operation is possible, as shown in Figure 5. In the decoupled case, a *Name Resolution System* (NRS) is used to map object names to locators that can be used to reach the corresponding information object, such as IP addresses. The NRS is some form of DHT, either a multilevel DHT [61] or a hierarchical SkipNet [62]. In the multilevel DHT solution, each authority maintains its own local NRS to handle the resolution of the `L` part, while a global NRS handles the resolution of the `A` part. A publisher makes an information object available by sending a PUBLISH message with its locator to the local NRS, which stores the `L` to locator mapping (arrow 1). The local NRS aggregates all the `L` parts for the same authority `A` into a Bloom filter [52], and sends a PUBLISH message to the global NRS (arrow 2). The global NRS stores the mapping between the authority `A` plus the Bloom filter and the local NRS, replacing any previous such mapping. When a subscriber is interested in an information object, it can send a GET message to its local NRS which consults the global NRS (arrows 3-4) in order to return a locator for the object (arrows 4-5). Finally, the subscriber sends a GET message to the publisher, using the returned locator (arrows 7-9), and the publisher responds with the information object in a DATA message (arrows 10-12).

In the coupled case, a routing protocol is used to advertise object names and populate the routing tables of *Content Routers* (CRs), as in NDN. A subscriber sends a GET message to its local CR, which propagates it hop-by-hop towards the publisher or a cache (arrows a-c). When the information object is found, it is returned via a DATA message, reversing the path taken by the GET message (arrows d-f). However, in contrast to NDN where pointers left in CRs are used for the return path, in SAIL the GET messages accumulate routing directions along their path, which are simply reversed at the publisher or cache in order to reach the subscriber.

In the hybrid mode of operation, the NRS returns *routing hints*, that is, partial locators that can direct a GET message in one or more directions where more information about the requested information object may be found. A GET message can thus start with some routing hints from the NRS to reach the vicinity of the requested information object, and then exploit name-based routing information stored in the CRs to reach its destination. Alternatively, a GET message can start with the name-based routing information stored in the CRs and resort to the NRS for further routing hints when a CR does not have sufficient information to forward it. As a result, routing in SAIL can be a mix of hop-by-hop and partial paths.

3) Caching: The SAIL architecture, in addition to on-path caching at the CRs, envisions the deployment of large scale information object caching and replication mechanisms in co-operation with the NRS, i.e., these caches are treated as publishers. SAIL considers a hierarchy of caches in which local caches are part of a tree that contains a small number of caching servers at the root. Caches higher up in the hierarchy have larger storage space in order to store popular objects, which otherwise would have been evicted by local caches due to their small size. The project also investigates cache migration policies, in which popular objects are dynamically migrated to caches that are closer to the consumers [60].

4) Mobility: Host mobility is supported by having the NRS maintain topological information for each registered

[5]While the architecture is most frequently referred to as NetInf, we use instead the project's name (SAIL) as for the other ICN efforts.

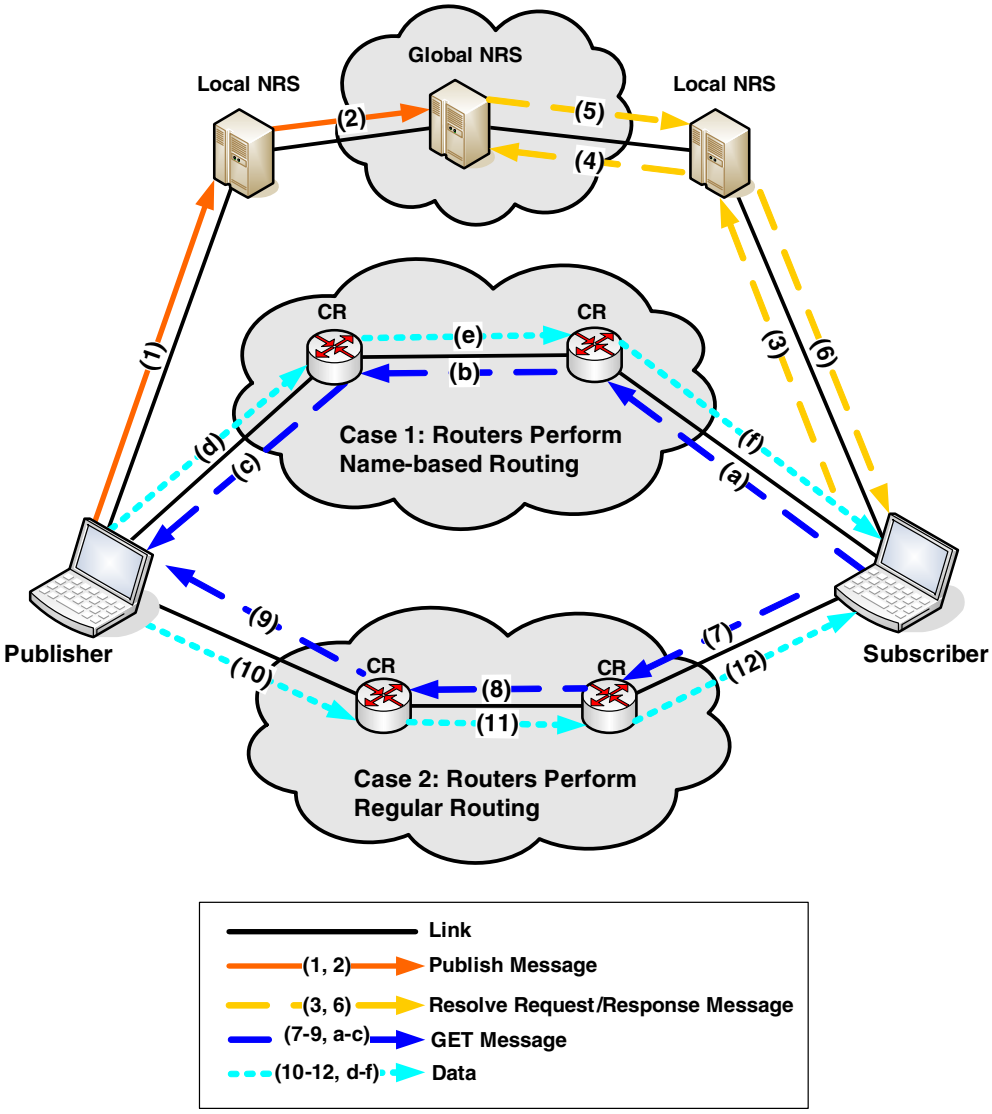

Fig. 5. The SAIL architecture. NRS stands for Name Resolution System, CR for Content Router.

host. Upon a change of location, the moving host updates the topological information in the NRS where it is registered and an appropriate notification is sent to any nodes that are currently communicating with the mobile host. The NRS also uses a *Late Name Binding* (LNB) strategy to allow the resolution process to terminate at a node close to the current area of a moving host. This is facilitated by the use of routing hints in the NRS, which allow messages to first be forwarded in the appropriate direction, and then get resolved to their exact destination. For example, when a publisher moves within its current area, it can simply update its local NRS with its location, without invalidating the routing hints in the global NRS that point to its current area.

5) Security: The SAIL architecture envisions a fully-fledged security system that covers name security, information integrity, authentication and confidentiality, and authorization and provenance. The basic building block of the security architecture is the inclusion of hash values in names, which allows self-certification of both the authority and the local part. SAIL names may explicitly identify the hash scheme used, e.g., SHA-1, to allow many such schemes to co-exist [60].

E. COMET

The *COntent Mediator architecture for content-aware nETworks* (COMET) [34] project (see Figure 1), funded by the EU Framework 7 Programme, is designing mechanisms for optimizing information source selection and distribution by mapping information to appropriate hosts or servers based on transmission requirements, user preferences, and network state [63]. The core component of the COMET architecture is a *Content Mediation Plane* (CMP) which mediates between the network providers and the information servers, being aware of both information and infrastructure. The COMET project has produced two very different architectures for the CMP: a coupled design called *Content-Ubiquitous Resolution and Delivery Infrastructure for Next Generation Services* (CURLING) [64], which is an ICN architecture with coupled name resolution and routing, and a decoupled

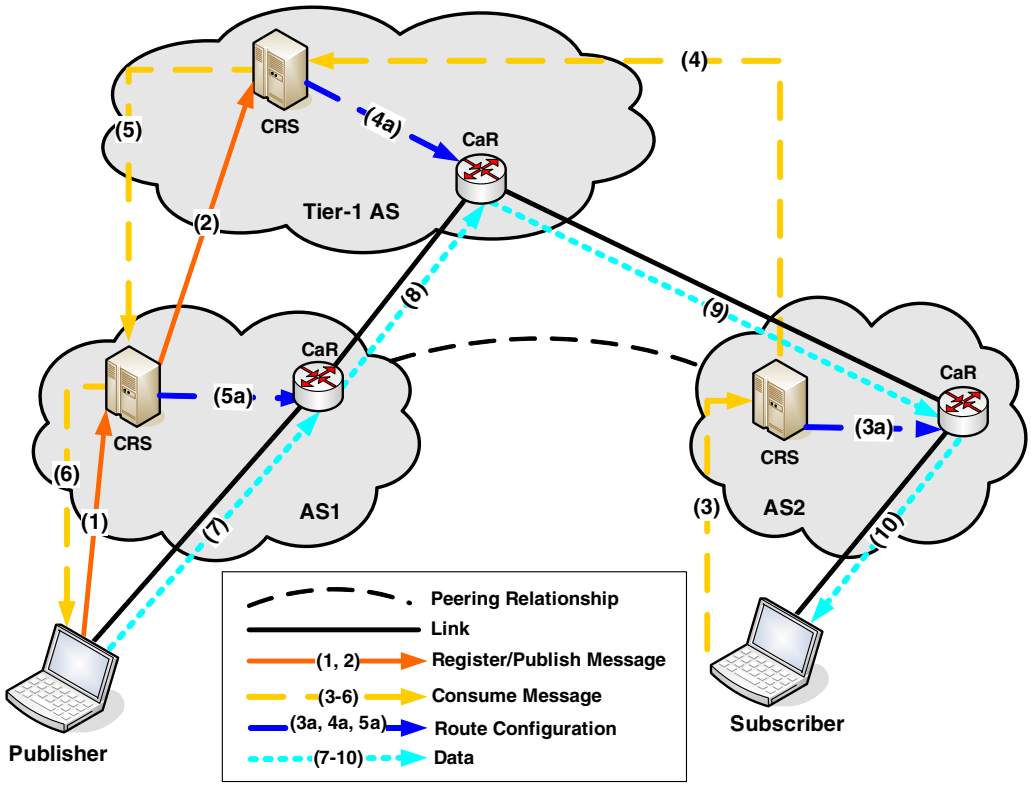

Fig. 6. The coupled COMET architecture. CRS stands for Content Resolution System, CaR for Content-aware Router.

design that enhances information delivery without fundamentally changing the underlying Internet [65]. Unlike other ICN approaches which strive for location independence, COMET allows both subscribers and publishers to explicitly include location preferences for information, following established business practices. For example, a subscriber may ask for bookstores in a specific country, and a publisher may only make videos available to a specific country.

1) Naming: A precise naming scheme has not been defined for COMET. However, in COMET the information names are provided by a *Content Resolution System* (CRS) when the information is registered by the publishers, thus allowing names for related information to be explicitly aggregatable, e.g., episodes of a TV series can have sequential names. This allows the naming system to scale by exploiting existing relationships between information objects.

2) Name Resolution and Data Routing: The coupled approach in COMET is presented in Figure 6. A publisher that wants to make some information available sends a REGISTER message to its local CRS node which issues a name for the information and stores the actual location of the information, e.g., the IP address of the publisher (arrow 1). This information is propagated upstream in the AS hierarchy using PUBLISH messages, so that each parent CRS ends up with a pointer to its child CRS that sent the PUBLISH message (arrow 2). The publisher may limit the propagation of this information to a specific area, e.g., an IP prefix, so PUBLISH messages may not reach the Tier-1 provider. A subscriber that is interested in some information issues a CONSUME message to its local CRS, which is similarly propagated upwards in the CRS hierarchy until it reaches a CRS that has information about that name (arrows 3-4). The subscriber may either limit the propagation of this information to a specific area or exclude specific areas from this propagation. When a match is found, the CONSUME message follows the pointers in the CRSs to reach the actual publisher (arrows 5-6). As the CONSUME message travels from the subscriber to the publisher, each CRS on the way installs forwarding state at the *Content-aware Routers* (CaRs) of each intermediate AS, pointing back towards the subscriber (arrows 3a-5a). The publisher can thus send the corresponding data to the subscriber by using these pointers (arrows 7-10).

While the coupled approach in COMET shares many ideas with DONA on name resolution and with NDN on data routing, there are some important differences. With respect to name resolution, in COMET the PUBLISH messages are not propagated to peering AS's, but only to parents, in order to reduce the state maintained at CRSs. This has two implications: first, when a CONSUME message reaches a tier-1 provider without finding a match, it must be propagated to all other Tier-1 providers to guarantee that a match will be found, if one exists, since all tier-1 providers are peers with each other; second, both name resolution and data routing (which are coupled) do not exploit peering links, therefore additional signaling is needed to switch to peering paths if available. For example, in Figure 6 data routing can switch to the peering link between AS1 and AS2 [64]. With respect to data routing, while in NDN both name resolution and data routing use the same CRs, in COMET name resolution uses the CRSs while data routing uses the CaRs, thus allowing the CRSs in each AS more flexibility in choosing the most appropriate paths between the available CaRs of that AS.

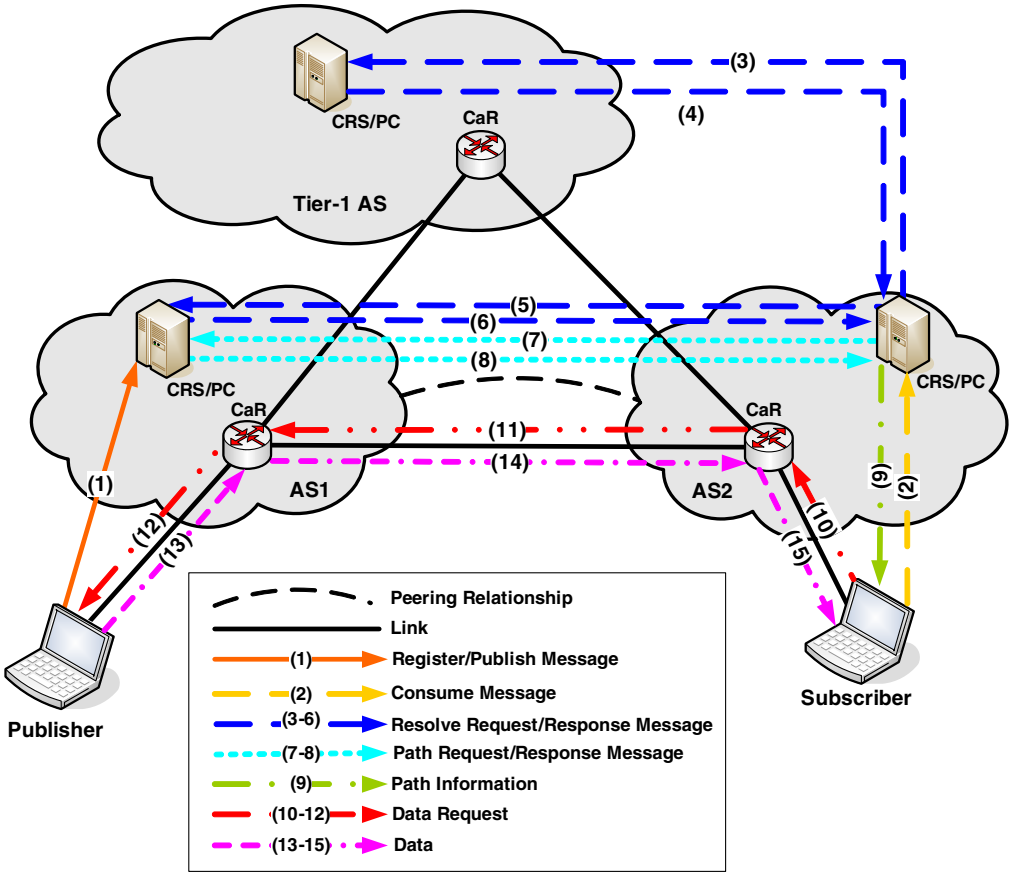

Fig. 7. The decoupled COMET architecture. CRS stands for Content Resolution System, CaR for Content-aware Router, PC for Path Configurator.

The decoupled approach in COMET is presented in Figure 7. In this case, the CRS system is similar to DNS, in that the CRSs split the object namespace among themselves in a fixed hierarchical manner. This means that when a publisher wants to make some information available, it simply sends a REGISTER message to its local CRS (arrow 1), which is not propagated further because it must belong to the namespace assigned to that CRS. When a subscriber issues a CONSUME message for some information (arrow 2), this is resolved by the root CRS to a pointer towards the publisher's CRS (arrows 3-4). The subscriber's CRS contacts the publisher's CRS to get the location of the publisher (arrows 5-6), e.g., its IP address. Then the subscriber's *Path Configurator* (PC) contacts the publisher's PC (shown co-located with the CRS nodes for simplicity) requesting a source route from the subscriber to the publisher (arrows 7-8). This source route is returned to the subscriber (arrow 9) which uses it to request information (arrows 10-12); its reverse is used by the publisher to return the information (arrows 13-15). COMET's decoupled approach has some of the limitations of DNS, for example, names are location-dependent due to the fixed assignment of namespace areas to network areas. As a result, it cannot be considered a true ICN architecture, hence in the remainder of this paper the term COMET architecture will refer only to the coupled approach.

3) Caching: COMET supports on-path and off-path caching in a manner similar to other coupled architectures, that is, on-path caching is a byproduct of name resolution, while off-path caching requires registering cached copies with the CRS. COMET has performed considerable work in on-path caching algorithms, proposing two novel schemes instead of NDN's "cache everything" scheme. ProbCache is a probabilistic caching scheme where each CaR first approximates how many times an information packet should be cached on a path by assuming other CaRs have the same caching capacity as itself and estimating the total traffic on the path by the requests it receives per unit time. Then, each CaR, based on this approximation, its distance from the publisher and its distance from the subscriber that made the request, probabilistically caches the packet [44]. On the other hand, the Centrality scheme is based on the observation that CaRs lying on many shortest paths are more likely to get a cache hit, hence an information object should only be cached by the CaR with the highest "centrality" in its path. The centrality is captured by the number of times a specific node is contained on the shortest paths between all pairs of nodes in a network topology. Computing the centrality at each CaR would require that every CaR has knowledge of the global topology. Thus, a simplified metric, for which a node needs only to know its 1-hop neighbors and the paths among them, is used [45]. Simulation results show that both caching schemes outperform the NDN scheme in terms of scalability and hit ratio [44], [45].

4) Mobility: COMET uses specialized mobility-aware CaRs placed at the edge of the access networks to support user mobility [66]. Mobility-aware CaRs track the mobility of users and information and can predict their future locations.

When a subscriber moves to a new CaR in the same domain, the latter can obtain the subscriber's context information from the previous CaR. When the subscriber moves to a different domain, the situation is more complicated since new CaRs may need to be configured with routing state after the handoff. As this process may lead to extended latencies during handover, a proactive handover approach can be adopted to avoid handoff latencies: the currently hosting domain predicts the domain to which the subscriber is likely to hand over, allowing for user context information to be transferred in advance.

5) Security: COMET adopts security techniques from other ICN architectures [65]. The security techniques that may be used depend however on the exact naming structure used. For example, if related pieces of information use sequential names for aggregation purposes, these names cannot use the self-certification approach of DONA which relies on embedding hashes in the names. One aspect of COMET that simplifies security provisioning is the use of AS paths rather than global addresses in both the CRSs and the CaRs, thus preventing attackers from using arbitrary network paths to launch undetected attacks.

F. CONVERGENCE

The CONVERGENCE [35] project (see Figure 1), funded by the EU Framework 7 Programme, envisions an ICN-based Future Internet that facilitates user access to information, spanning from digital data and services to people and real-world objects. Each such object in CONVERGENCE is represented by a *Versatile Digital Item* (VDI), a common container for all kinds of digital information, based on the MPEG-21 specification. A *Content Network* (CONET) [67] allows publishers to make available VDIs and subscribers to express interest in those VDIs. A distinguishing characteristic of the CONVERGENCE architecture[6] is that it attempts to ease transition from IP by reusing existing functionality. For example, since CONVERGENCE messages are expected to be large due to naming and security meta-data, rules are defined for splitting them to *carrier packets*, e.g., IP datagrams. Furthermore, an IP header option has been defined to carry the essential information from CONVERGENCE message headers, allowing CONVERGENCE-aware IP routers to treat IP datagrams containing CONVERGENCE messages differently.

1) Naming: In CONVERGENCE object names consist of a `namespace ID` and a `name` part, whose format is determined by the `namespace ID`. While the default format of CONVERGENCE names is similar to that in DONA, i.e., a flat `P:L` pair [67], hierarchical names may also be used as in NDN [68], or even URLs. The exact properties of the names depend therefore on the specific namespace used. Since CONVERGENCE is most similar to NDN, we assume in the following the use of hierarchical names.

2) Name Resolution and Data Routing: The CONVERGENCE architecture, shown in Figure 8, has many similarities with NDN; indeed, its prototype has been implemented as a modification of the NDN prototype [68]. Subscribers issue INTEREST messages requesting an information object, which are forwarded hop-by-hop by *Border Nodes* (BNs) to publishers or *Internal Nodes* (INs) that provide caching (arrows 1-3 and 6). Publishers respond with DATA messages which follow the reverse path (arrows 7-10). In order to reduce the state requirements at the BNs, CONET diverges from NDN in three aspects. First, BNs do not maintain name-based routing information for every advertised name prefix, but only for a small portion of them, hence their routing table operates like a route cache. If an INTEREST message cannot be forwarded because there is no routing information for the corresponding name, the BN consults an external *Name Resolution System* (NRS), e.g., DNS, in order to find out how to forward the INTEREST (arrows 4-5). Second, as INTEREST messages are propagated they accumulate the network addresses of the BNs they pass, allowing the publisher to route the DATA message by reversing this path information, without requiring the maintenance of pointers at BNs. Third, BNs do not have to be directly connected; instead, the path between two BNs can involve multiple hops, e.g., via IP routers as shown in Figure 8, hence their designation as *border* nodes. Therefore, unlike CRs in NDN, BNs map names to network addresses, e.g., IP addresses, rather than to interfaces.

In CONVERGENCE name resolution and data routing are coupled, since the path taken by a DATA message is the reverse of the path followed by the corresponding INTEREST message, even though each step of this path may not be a single hop but an entire IP path, hence the path segments between BNs which an INTEREST message and its corresponding DATA message follow are not necessarily symmetric. The NRS is used if an appropriate route is not found at some BN. The details of the NRS used have not been defined by the CONVERGENCE project. The name-based routing tables at BNs may also be partially populated without resorting to the NRS, by running a routing protocol for name prefixes, e.g., OSPF, as in NDN.

3) Caching: CONVERGENCE supports on-path caching in a manner similar to NDN. Off-path caching and replication are supported by registering additional copies of an information object stored at INs to the NRS; however, the signaling overhead for this registration is unclear, as an NRS mechanism has not been defined yet for CONVERGENCE.

4) Mobility: In principle, CONVERGENCE supports subscriber mobility exactly as in NDN, i.e., by having subscribers issue new INTEREST messages after a handoff. However, the efficiency of CONVERGENCE in supporting subscriber mobility is questionable, as it decouples forwarding information from BNs. More specifically, in NDN the re-issued INTEREST messages are suppressed when they reach the first BN that had received a previous such message. This does not apply to CONVERGENCE, since BNs do not maintain per INTEREST state. Therefore, pre-handoff and post-handoff INTEREST messages are treated separately by the network, propagating all the way to the publisher, where they trigger independent (and duplicated) DATA messages in response. Publisher mobility on the other hand requires updating the NRS, whose overhead is unknown, as noted above.

5) Security: CONVERGENCE adopts the per DATA message security approach of NDN, i.e., each DATA message

[6]While the architecture is most frequently referred to as CONET, we use again the project's name (CONVERGENCE) as for the other ICN efforts.

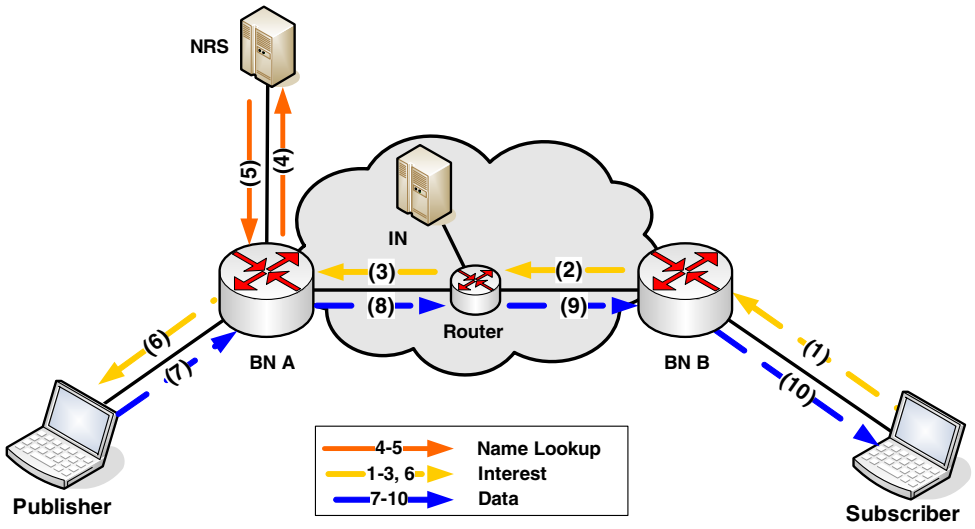

Fig. 8. The CONVERGENCE architecture. NRS stands for Name Resolution System, BN for Border Node, IN for Internal Node.

contains a digital signature. Due to the large overhead of the meta-data required for signature verification, DATA messages are expected to be much larger than carrier packets, e.g., IP datagrams encapsulated in Ethernet frames. For this reason, CONVERGENCE proposes performing security checks on information only at the DATA message level at the subscriber, leaving the BNs to only deal with the (smaller) carrier packets that cannot be individually authenticated [68].

G. MobilityFirst

The *MobilityFirst* [38] project (see Figure 1), funded by the US Future Internet Architecture program, proposes a clean-slate Future Internet architecture with an emphasis on treating mobile devices as first-class citizens [69]. As a result, MobilityFirst provides detailed mechanisms to handle both mobility and wireless links, as well as multicast, multi-homing, in-network caching and security. The basis of the MobilityFirst architecture is the separation of names for *all* entities attached to the network (including information objects, devices and services) from their network addresses: each entity has a globally unique name, which can be translated into one or more network addresses at various points in the network, thus allowing messages to be dynamically redirected in order to follow a mobile device or content.

1) Naming: Each network entity in MobilityFirst is assigned a *Globally Unique Identifier* (GUID) via a global naming service that translates human-readable names to GUIDs. Every device in MobilityFirst must obtain GUIDs for itself, its information objects, and its services. GUIDs are flat 160-bit strings with no semantic structure and they may be randomly selected, since their length ensures that the probability of a collision is small. Alternatively, GUIDs can be self-certifying hashes of information objects, thus allowing information integrity verification, or hashes of public keys, thus binding devices to principals. Each network attached entity has a unique GUID, and if an entity (e.g., video file) is available in multiple network locations, then all of its copies will have the same GUID. By naming all network entities, MobilityFirst can support both name-based information delivery (via information GUIDs) and host-to-host communication (via device GUIDs).

2) Name Resolution and Data Routing: In MobilityFirst all communication starts with GUIDs, which are translated to network addresses in one or more steps, via a *Global Name Resolution Service* (GNRS) as shown in Figure 9. A publisher that wishes to make some information available asks the naming service for a GUID and then registers the GUID with its network address in the GNRS (arrow 1). A GUID is mapped via hashing to a set of GNRS server addresses, which are contacted using regular routing [70]. When a subscriber wants to receive some information, it sends a GET message that includes the GUID of the requested object, along with its own GUID for the response, to its local *Content Router* (CR) (arrow 2). The CR can only route based on actual network addresses, e.g., IP addresses, hence it asks the GNRS for a mapping between the destination GUID and one or more network addresses (arrow 3). The GNRS replies (arrow 4) with a set of network addresses (optionally it may also send a source route, a partial source route and/or intermediate network addresses). The CR selects one of these network addresses, adds it to the GET message, which it then forwards using the regular routing tables in the CRs (arrows 5-6 and 9). The GET message includes both the destination GUID and the destination network address, and any CR along the path can consult the GNRS to receive an updated list of network addresses for the destination GUID (arrows 7-8) if, for example, due to mobility the GET message cannot be delivered to the publisher. The publisher sends its response to the subscriber's GUID, using the same procedure (arrows 10-13).

The resulting name resolution and data routing process is a hybrid between IP routing and name-based routing. The actual routing is performed based on network addresses, with the GNRS only used to map GUIDs to network addresses. For less dynamic services, MobilityFirst can translate each GUID to a network address once, as with DNS, and operate based on network addresses only, ignoring the GUID. For more dynamic

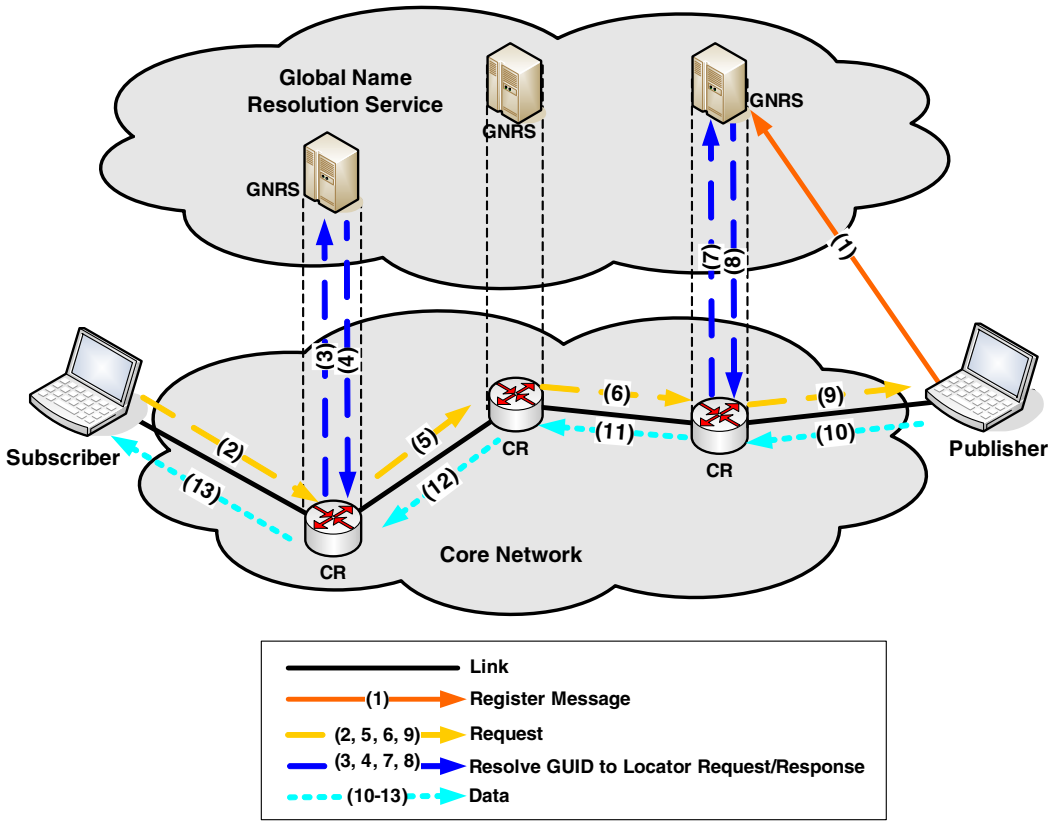

Fig. 9. The MobilityFirst architecture. GNRS stands for Global Name Resolution Service, CR for Content Router.

services, the GUID may be translated multiple times; the first router (optionally, others too) asks the GNRS for the network addresses bound to a given GUID and makes forwarding decisions based on the reply from the GNRS. Forwarding can thus be "fast path", when the GNRS is bypassed, or "slow path", when routers (re-)consult the GNRS in order to obtain an updated list of network addresses. This late binding or re-binding is especially useful for mobile destinations. Note that each message is delivered separately, i.e., the GET message and the information object sent in response to it are individually routed based on their destination GUIDs, therefore name resolution and data routing are decoupled in MobilityFirst.

3) Caching: MobilityFirst supports on-path caching by opportunistically caching passing messages at intermediate CRs, thus allowing subsequent requests for the same GUID to be answered with the locally cached copy. In addition, each time an information object is cached off-path or replicated, the GNRS is informed of the change in order to update the corresponding GUID entry with the additional network addresses. While the GNRS can be repeatedly consulted as a message travels through the network, this is a "slow-path" operation, and each CR can implement its own policy on when to consult with the GNRS for additional cached copies.

4) Mobility: In MobilityFirst the aim is to address host, information, and entire network mobility. Host mobility is primarily handled by the GNRS, which must be updated when a network attached object changes its point of attachment. No routing level indirections such as those performed in mobile IP are required. Network mobility is supported at lower levels and another distributed protocol (analogous to BGP) can be utilized for disseminating routing updates. While BGP can be used for inter-domain routing, a storage-aware routing mechanism can be employed at the intra-domain level in order to best support networks where disconnections due to mobility and variable link conditions are prevalent, by exploiting local storage, as in delay-tolerant networking [71].

5) Security: MobilityFirst envisions a decentralized trust model for name certification, where independent naming organizations exist (e.g., one per country or one per institute) for mapping human-readable names to GUIDs. The GUID of an entity can be securely bound to that entity via cryptographic techniques, thus enabling traffic accountability. On the other hand, users can frequently request a new GUID to avert profiling.

IV. Comparison of Approaches

The ICN architectures discussed in Section III have both similarities and differences in the way they implement the key ICN functionalities. In addition to summarizing the most salient characteristics of each ICN architecture in Table I, in this section we provide a more detailed analysis and comparison of the different design choices.

A. Naming

The naming of information objects and services is one of the fundamental design choices in each ICN architecture, since the structure of names and their semantics have a profound impact on all other aspects of the architecture. The main choices are

TABLE I
IMPLEMENTATION OF KEY FUNCTIONALITIES IN ICN ARCHITECTURES.

	Naming	Name Resolution and Data Routing	Caching	Mobility	Security
DONA	Flat, consisting of principal and label part.	Resolution handlers organized following AS hierarchy. Coupled: A source-route is created during resolution. Decoupled: Resolution handlers return network address.	On-path caching at resolution handlers. Off-path caching requires additional registrations.	Subscriber mobility via new requests. Publisher mobility requires additional registrations.	Principal is hash of public key. Label can be hash of immutable content.
NDN	Hierarchical, may contain publisher-specific prefix.	Routing protocol used to flood name prefix information. Coupled: Routing state for data is established at routers during request propagation.	On-path caching at content routers. Off-path caching requires additional routing information.	Subscriber mobility via new requests. Interest flooding protocol for publisher mobility.	Signatures included in packets. Certification chain can follow name hierarchy.
PURSUIT	Flat, consisting of scope and rendezvous part. Scopes may be organized hierarchically.	DHT-based rendezvous network matches subscriptions to publications. Scopes can be used to limit publication/resolution. Decoupled: topology management and forwarding are separated from rendezvous.	On-path caching difficult due to decoupled operation. Off-path caching requires additional registrations.	Subscriber mobility via new requests. Publisher mobility requires updating the topology manager.	Packet level authentication for individual packets. Names can be optionally self-certifying.
SAIL	Flat-ish names, consisting of authority and local part. Possible name aggregation for the same authority.	Coupled: requests accumulate routing state during name resolution. Decoupled: DHT-based name resolution returns content locator. Hybrid: name resolution returns routing hints to assist coupled operation.	On-path caching at content routers. Off-path caching requires additional routing information or registrations.	Subscriber mobility via new requests. Support for publisher mobility via routing hints in hybrid operation.	Names can be optionally self-certifying.
COMET	Unspecified. Names produced by name resolution system which creates aggregatable names for related content.	Coupled: DONA-like resolution but forwarding state is installed in content routers during resolution. Scoping/filtering used to limit publication/resolution.	Probabilistic on-path caching at content routers. Off-path caching requires additional registrations.	Specialized mobility-aware content routers at access network that exchange mobile context state.	Unspecified, but explicitly aggregatable names make self-certification impossible.
CONVERGENCE	Either flat or hierarchical, most work based on hierarchical.	Name prefixes cached at content routers, unspecified name resolution system for non-cached prefixes. Coupled: Interest messages accumulate routing state during name resolution.	On-path caching at content routers. Off-path caching relies on unspecified name resolution system.	Subscriber mobility via new requests. Publisher mobility relies on unspecified name resolution system.	Signatures included in objects larger than packets.
MobilityFirst	Flat, consisting of a single component.	Hash –based global name resolution service to map names to network addresses, may be used repeatedly for late binding of addresses. Decoupled: requests and data are independently resolved and routed.	On-path caching at content routers. Off-path caching requires additional registrations.	Subscriber and publisher mobility via late binding to first reach the mobile's area and then find the mobile.	Names can be optionally self-certifying.

hierarchical or flat names, and there is an ongoing debate on the merits of each approach [40].

The main argument for hierarchical names is that they allow the system to scale via aggregation. This is an especially critical issue if we consider the number of named objects that will need to be supported, which are estimated between 10^{13} and 10^{15} [61], which is way beyond what current systems, e.g., DNS or BGP, can handle. NDN, which uses hierarchical names for all resolution and routing decisions, assumes that extensive name aggregation will be applied in content routers (especially at the core) in order to reduce routing table sizes. However, to achieve significant name aggregation, names must be allocated in a manner that reflects the actual network topology. For example, consider content router A in Figure 3. Its routing entries indicate that all names with the prefix `/aueb.gr/` are reachable via content router C, with the exception of the `/aueb.gr/cs` prefix which is reachable via content router B. Assume now that a new prefix `/aueb.gr/db` is announced to content router A. If it is reachable via content router C, then it will be aggregated with the existing entry for `/aueb.gr/`, but if it is reachable via content router B, then a new routing entry will be required, as it cannot be aggregated with the entry for `/aueb.gr/cs`. This shows that either *all* names with the same prefix must point at the same area, or the prefix will not be aggregatable. Unfortunately, binding names to locations has been identified as one of the main shortcomings of the current Internet architecture with respect to mobility support, and there are several pre-ICN proposals for decoupling object names from topological addresses [72], [73], [74]. Therefore, the exact benefits of hierarchical name aggregation to scalability are debatable.

The main argument for flat names is that they avoid this location-identity binding, thus simplifying mobility. However, flat names are not easy to aggregate, thus requiring either huge routing and/or name resolution tables, or complicated and costly solutions like DHTs, as discussed in the following section. In addition, these data structures also need to be updated whenever information moves.

Another aspect of the debate between hierarchical and flat names is that the former can be human-readable while the latter can be self-certifying; unfortunately, one cannot have both [40]. For example, NDN has to rely on an external trust mechanism to bind signed information to human-readable names [47], while MobilityFirst has to rely on an external naming system to bind human-readable names to GUIDs [69].

B. Name Resolution and Data Routing

For ICN architectures based on hierarchical names, the main issue in name resolution is how to scale the system without requiring location-identity binding. Initial work on NDN suggests using ISP-provided names as prefixes for achieving name-aggregation, since there is yet no clear way to avoid this without losing the scalability benefits of hierarchical aggregation [75]. CONVERGENCE tries to bypass the scalability problem by caching only a limited number of name prefixes in the routing tables, but this requires an external name resolution system, which would face the same scalability problems.

For ICN architectures based on flat names on the other hand, the main issue in name resolution is how to handle a huge naming space that cannot be aggregated. In DONA and COMET name resolution state is accumulated as we move towards tier-1 ISPs, requiring the top-level resolution servers to store huge amounts of data, especially in DONA which replicates everything at the top level [51]. COMET tries to partially mitigate this by using scoping to limit the propagation of name resolution information; it also proposes creating explicitly aggregatable names to reduce state requirements, but these names cannot be self-certifying. PURSUIT relies on a hierarchical DHT to spread this load, at the cost of inflating name resolution paths, routing policy violations and reliance on external name resolution servers for local information [50], [51]. SAIL relies on a two level (local and global) DHT solution in order to always locally resolve requests for global information, at the cost however of binding names to specific AS's [61]. MobilityFirst distributes the name resolution service by using a hashing scheme to determine the address of a name resolution server for each GUID, relying then on regular routing to reach this server; this however requires that all AS's in the world implement the same scheme [70].

Another proposed approach is to explicitly aggregate flat names [40]. In this approach, a request contains a series of concatenated flat names, e.g., `A.B.C`, and routing decisions are based on a *deepest match*: the router searches for matches in its routing table from right to left. In the above example, a router would first look up `C`, if that failed it would look up `B`, and so on. Explicit aggregation allows reducing the size of routing tables, e.g., an entry for `A` is enough for addressing requests for `A.B.C`, as long as users include `A` as a routing hint in their request.

The approaches to data routing taken by ICN architectures have already been classified as coupled or decoupled from name resolution. In the coupled approaches, routing information is accumulated during name resolution, therefore data are normally forwarded by reversing the path taken by requests, while in the decoupled approaches the request and data paths may be totally different. The exact details however vary greatly between ICN approaches. Among the coupled approaches, NDN and COMET install forwarding information in content routers as requests are resolved towards the publisher, while CONVERGENCE and the coupled variants of DONA and SAIL accumulate this information in the request packets themselves and then rely on source routing to return the corresponding data to the subscribers. As a result, in the first case the data routing state is maintained in the content routers, while in the second case the data routing state is included in the data packets themselves.

Among the decoupled approaches, the decoupled versions of DONA and SAIL rely on the name resolution system to return an IP address for the desired information, which can then be reached via regular IP routing; multiple addresses may be returned if the information is available at many locations. PURSUIT uses an independent topology management entity to calculate data routing paths, which are encoded in Bloom filters that are used as source routes; Bloom filters however do not scale to Internet sizes and suffer from false positives as more links are added to them [53]. MobilityFirst relies on a

very fast name resolution system to iteratively retrieve network addresses [70], starting with general directions and ending with specific addresses, so as to best support mobility. In addition, MobilityFirst treats requests and data in a symmetric manner, thus simplifying subscriber *and* publisher mobility. We note that, independent of the exact details of how name resolution and data routing are performed, decoupling name resolution and data routing allows more flexibility in how each function is implemented, allowing, for example, different paths to be used for signaling (control) traffic and data traffic.

C. Caching

Caching is a fundamental feature of ICN architectures, as information awareness allows the network to identify cached information without resorting to the application layer, as in Web caching. In on-path caching, when a router receives a request for a piece of information it responds with a locally cached copy, without involving the name resolution system. In off-path caching, caches announce their information to the name resolution system, so that they may be matched to information requests that would not normally reach them, essentially becoming alternative information publishers. On-path caching is generally opportunistic, i.e., routers cache information that happens to flow through them, while off-path caching can also be used to actively replicate information, as in CDNs.

While all ICN architectures natively support on-path caching in principle, when name resolution and data routing are decoupled there are less opportunities to exploit opportunistic caching, as the name resolution path generally differs from the data routing path: while the information can be opportunistically cached on the data routing path, subsequent requests for the same information follow the (different) name resolution path, reducing the possibility for a cache hit. When name resolution and data routing are coupled on the other hand, if data is cached on the data routing path it will result in a cache hit when subsequently requested over the same name resolution path. Opportunistic caching can range from the "cache everything" approach of NDN, to the probabilistic caching approach of COMET [44], [45].

In off-path caching (and replication), beyond the more general problem of choosing what to cache and where [55], the main issue is how to reduce the overhead required in order to inform the name resolution system when new items are cached or old items are discarded. The exact details depend on the name resolution scheme used, but one common goal is to keep updates local, e.g., within an AS, so as to reduce signaling overhead and only serve customers from within that AS. In DONA and COMET cached information can be advertised only within an AS and not propagated upwards in the AS hierarchy (COMET provides the scope mechanism for this purpose). Similarly, in the decoupled version of SAIL and in PURSUIT cached information can only be advertised within the local DHT of an AS. In NDN, CONVERGENCE and the coupled version of SAIL, the name prefix tables need to be updated, but it is unclear how this could be achieved economically, as the routing protocols proposed for advertising name prefixes are based on flooding. MobilityFirst also faces problems in this area, as it relies on a global lookup mechanism for name resolution, therefore it is unclear how locally cached copies can be advertised only within an AS.

D. Mobility

The location-identity split in ICN approaches and the stateless nature of the publish/subscribe paradigm can potentially facilitate the support of mobility. In practice however, the situation is not that simple. In some approaches the location-identity split may be partial for scalability purposes, e.g., in NDN; in others, the cost of looking up network addresses may be too high for fast moving objects, e.g., in DONA. Furthermore, while individual requests for information objects can be issued from different locations, mobility during the reception of a simple object remains a problem, something more likely to occur with larger information objects.

Supporting subscriber mobility is generally simpler. In the worst case, the subscriber will just issue new requests for information, wasting any resources spent on pending transmissions. This is the most sensible approach in the decoupled versions of DONA and SAIL, where the name resolution system returns the network addresses of the host carrying the requested content. In NDN and COMET on the other hand, re-issued requests will eventually cross paths with the state left in content routers by the old requests, thus redirecting the pending information to the new location of the mobile subscriber. ICN architectures based on source routing require routes to be patched after a mobile moves, to point at its new location. This may not be very efficient, but it is relatively easy in CONVERGENCE and the coupled variants of SAIL and DONA where each part of the path is visible. It is trickier in PURSUIT where links are encoded into a Bloom filter, as it is easy to add links to the Bloom filter, but hard to remove them. MobilityFirst offers the most flexible solution, as it relies on a series of resolution steps to delay binding a mobile's identity with its current address as much as possible, thus allowing mobiles to only update their location in their local area.

Publisher mobility requires updating the name resolution system with the new location of the mobile publisher. Unlike off-path caching where we would prefer caches to only be advertised locally, with publisher mobility the information needs to be globally available. While updating the name resolution system can be a costly operation, it can be kept within the local area in systems supporting late binding of information to location, such as MobilityFirst and the hybrid variant of SAIL. Note that the hybrid variant of SAIL only applies late name binding to requests, not data, hence it cannot be used to handle subscriber mobility.

E. Security

All ICN architectures identify security as a fundamental problem of the current Internet architecture. The primary focus of the proposed architectures is on information confidentiality and integrity, as opposed to the channel confidentiality and integrity of IP based solutions. Content confidentiality and integrity are achieved by using cryptographic mechanisms either combined with data included in the information name,

as in DONA, or associated with the information using meta-data, as in NDN. As we explained above, each approach requires an external trusted system to complete the binding between a human-readable name, a name for the information and the information itself. Most ICN architectures rely on self-certifying names, as these allow any network node to verify that the name in a packet matches the information inside it, leaving to the user the problem of determining whether this information is actually what was desired. While both hierarchical and flat names can be transformed to self-certifying names, this process is easier when the latter are used, hence all systems supporting flat names generally allow self-certifying names to be used.

V. Open Issues in ICN

In this section we identify a series of issues and problems that have either not been satisfactorily addressed or have not even been tackled by the ICN research community so far.

A. Naming

There is no clear consensus yet on whether hierarchical or flat names should be used. Hierarchical names can be human-readable and are easier to aggregate in principle, but it is unclear whether they can scale to Internet levels without turning into DNS names due to aggregation. On the other hand, flat names are easier to administer, they do not impose processing requirements for longest prefix matching, they can be self-certifying and they can be easily handled with highly scalable structures such as DHTs, but it is unclear whether DHTs can offer satisfactory performance.

There has been practically no research on incorporating versioning, deletion and revocation of information objects to the naming structure, and only preliminary work on the optimal granularity of information objects (i.e., an object could correspond to a packet, to variable-sized information chunks or to entire application-level objects). Indeed, some work argues that performing signature checks on individual packets may have excessive overhead [68], while other work argues that this is feasible with hardware-level implementations [57].

Searching for information has also not received much attention in ICN research, something rather peculiar, given that most projects rely on flat names that have to be somehow discovered by human users. Information-awareness may provide the means for efficient searching, possibly taking into account meta-information such as contextual parameters, location, information type, language, etc. For example, SAIL envisions an extended name resolution system that integrates meta-information to the resolution process [60]. As information is the primary entity in ICNs, it is possible for this meta-information to co-exist with the actual information inside the network, thus allowing the intelligent manipulation of traffic for other purposes, such as for enabling geocasting and flow prioritization. However, the availability of such meta-information also raises significant concerns regarding network neutrality. Earlier attempts to throttle certain types of traffic (e.g., P2P) were based on DPI techniques. With ICN, the identification of traffic types (and of any other meta-information related to a flow) may constitute standard network functionality, thus unveiling sensitive information not only to ISPs, but also to potential attackers.

B. Name Resolution

The vast size of the naming space poses a significant scalability challenge for name resolution. DHT based designs have attracted the attention of researchers due to their logarithmic scalability. The routing policy violations and inflated path lengths of DHTs have resulted in hierarchical schemes that try to adapt the structure of the name space to the underlying inter-domain network topology [76], but the routing efficiency of these approaches is still lacking [51]. Moreover, recent studies on the structure of the inter-domain graph suggest that the increase of peering relationships between AS's gradually leads to a mesh-like inter-domain graph [77], [78], therefore, employing a strictly hierarchical structure for the organization of the name space does not seem to reflect reality. Another recently proposed approach is to use hashing to map names directly to IP addresses and rely on IP routing to find the resolvers [70], but this requires global participation in the name resolution system. Hence, a flexible and practical approach, able to express the dynamically evolving routing relationships between AS's, is still lacking.

C. Data Routing

While a lot of effort has been devoted to the design of routing mechanisms for the intra-domain level, e.g., [53], little attention has been paid to the inter-domain level. Inter-domain routing is strongly affected by business relationships between the involved parties and is an area of active research even in the context of the current Internet architecture [79]. In the ICN area, the main issue is scaling the proposed solutions to Internet sizes. As shown in [80], the content routers in NDN face serious scalability limitations at the inter-domain level, something that also applies to some extent to COMET, which also installs forwarding state at routers.

In the PURSUIT architecture which uses in-packet Bloom filters for source routing, the most obvious issue is that longer paths (or larger multicast trees) lead to many false positives, i.e., wasted packet transmissions [53]. Since larger Bloom filters would introduce much higher overhead, ideas such as Bloom filter switching [81] and variable-sized Bloom filters [82] have been explored. But the real problem is establishing inter-domain paths, since it is unrealistic to expect topology managers to have a global view of the network, due to both the size of the Internet and the limited information exchanged between AS's. This means that a hierarchical decomposition of the inter-domain routing problem is required, coupled with Bloom filter switching between the AS's, to keep topology management local and path lengths short [81], [83].

On the other hand, in the architectures where source routes are accumulated during name resolution, such as CONVERGENCE and the coupled variants of DONA and SAIL, the main issue is the amount of overhead introduced in both request and data packets as these routes grow larger. MobilityFirst and the decoupled variants of DONA and SAIL basically rely on IP routing, with the possibility of additional resolution steps in MobilityFirst and the hybrid variant of SAIL. This

means that they do not introduce any new problems, but they, at least partially, inherit the existing problems of IP routing.

D. Caching

Mechanisms for caching (and replication) have been widely studied at the application level, mostly in the context of web applications. It has been recently advocated that the benefits from the extensive use of caching in ICN will not be substantial [27]. Although they raise serious concerns about the performance of the envisioned caching mechanisms, these observations are mostly based on studies performed more than a decade ago [84]. Additional research on current traffic patterns could shed additional light on the popularity characteristics of information today and thus to the possible benefits from widespread caching. For instance, a recent study has shown that web information popularity has changed during the past few years, affecting application level caching performance [85].

Another issue is that when caching takes place inside the network, as in ICN, several types of traffic will compete for the same caching space. Cache space management therefore becomes crucial for the network, and recent works, albeit based on simplified traffic models, have indicated that intelligent schemes can substantially improve performance [45], [44], [86]. Moreover, the deployment of caching and replication mechanisms inside the network opens up the possibility of jointly optimizing routing, forwarding and in-network cache management. For instance, routing decisions could be affected by cache locations, the cache-ability of information and/or indications of cache contention.

E. Mobility

Though identified as a major shortcoming of the current Internet architecture, network support for mobility has received very limited attention in ICN efforts (e.g., [46], [56]). Past research efforts on the support of mobility in the context of publish/subscribe systems [28] and on multicast-assisted mobility [87] have contributed to the understanding of the emerging issues. This work, coupled with the native ICN support for caching and multicast, has been leveraged to assist mobility in PURSUIT [54]. However, publisher (and, therefore, information) mobility remains a major challenge, since most ICN architectures use name resolution systems that are slow to update, whether they are name-based routing tables, hierarchical DHTs or hierarchical resolution handlers. The use of source routes, that may become invalid even as they are formed, is an additional complication. Even more problematic is the use of name aggregation in routing tables, as it implicitly reintroduces a location-identity binding. The most promising approach in this area is the late name binding advocated by MobilityFirst and the hybrid variant of SAIL, which simplify mobility management without losing the advantages of flat names. The performance of these schemes in practical and large scale scenarios remains to be seen, however.

F. Security, Privacy and Trust

Security in all ICN architectures is based on using encryption with keys associated with the information name. Little work exists however on how these keys will be managed, i.e., who will be responsible for creating, distributing and revoking those keys. The need for key management mechanisms becomes of paramount importance if we consider the fact that most ICN approaches rely on cryptographic keys and trusted entities for information-name verification [40], [47]. Moreover, most of the proposed ICN architectures envision access control mechanisms, nevertheless there is very little work on the definition of access control policies, the application of access control policies to cached information and the authentication of users (e.g., [88], [89]).

ICN architectures can create severe privacy threats, as users reveal their interest in particular information and the name of the information being requested is available to all the ICN nodes processing the request [27]. A convincing solution for this threat has not been provided yet. Finally, efficient mechanisms for building trust relationships and handling privacy tussles amongst the various stakeholders are envisioned in ICN architectures (e.g., [90]), yet this still remains an open issue.

G. Transport

The information awareness in ICN architectures enables a series of new mechanisms and functionalities inside the network that make data transport a more complicated process than in the current end-to-end model. Mechanisms such as in-network caching and replication offer the opportunity for exchanging bandwidth with storage, thus radically changing the transport layer. Moreover, new delivery modes such as multicast (i.e., one-to-many) and concast (many-to-one), the ability of the network to apply anycast, as well as the support for multi-path routing in several ICN approaches, offer a rich set of mechanisms affecting the design of flow, congestion and error control functions. However, the fact that ICN architectures are still under active development, complicates research in the area. Recent efforts have started to investigate the interaction of these mechanisms (e.g., [91], [92], [93]), which is however far from being well-understood.

H. Quality of Service

Most ICN initiatives devote some thought to *Quality of Service* (QoS) provisioning. Nonetheless, only a few of them provide details about practical QoS mechanisms, while the rest treat the issue superficially. The most extensive treatment of QoS issues is in the COMET architecture which defines three *Classes of Service* (CoS) used to prioritize end-to-end information traffic. COMET maps the delivery requirements of the information as expressed by a CoS into the network paths offered by each AS via a path provisiong process [65]. Some work has also been performed on exploiting the centralized topology management and source routing of PURSUIT to implement routing algorithms that are infeasible with distributed routing, such as Steiner tree-based multicasting [94].

I. Business and Deployment Aspects

Taking a step away from technical issues, a series of questions need to be answered with regard to the business aspects of ICN. To name but a few: Who are the new actors

enabled by ICN architectures? How are the roles/relationships between current actors of the Internet ecosystem going to be affected? Which are the application domains to target first? Should overlay or native ICN solutions be deployed first? For example, CDNs already provide several features of the ICN paradigm at an overlay level. It is not clear however how CDNs would possibly fit in an ICN world, as a major part of their functionality would be provided by the network itself.

A first attempt to perform a socio-economic analysis of an ICN architecture was performed in the PSIRP project [95]. According to its findings the logical order of markets to target would be government, business ICT, and information-centric applications. This is because the business opportunities in the government sector can be satisfied with the adoption of purely overlay mechanisms, which entail a smaller overall cost compared to the adoption of native mechanisms. On the other hand, native mechanisms are necessary to fully exploit the business opportunities related to the business ICT sector. Finally, the investment in information-centric applications is strongly dependent on traffic volumes, which in turn depend on the widespread access to applications, and hence requires a widespread deployment of the new architecture. According to the same analysis, the adoption of an ICN architecture should start with the adoption of overlay mechanisms in the current Internet, followed by the adoption of native mechanisms on the network backbone. The adoption of such native mechanisms should start from the business ICT sector. Issues like billing, costing and invoicing for ICN traffic however remain open.

With regard to deployment, it is clear that an incremental transition into ICN is needed, so as to maintain compatibility with TCP/IP-based applications for an extended period. Although such a transition is straightforward for overlay ICN solutions, it is not well understood how it can be achieved for the case of clean-slate ICN solutions. In addition the ICN community has not reached a consensus on several fundamental design choices (e.g., routing and forwarding in NDN vs. PURSUIT) hence there are several architectures proposed, each fitting the requirements of different networking environments and/or business scenarios. It is therefore possible to reach a state where multiple different ICN architectures will be deployed in parallel and interoperability issues may arise.

VI. Conclusions

We have attempted to provide an in depth survey of the ICN research landscape. As a first step, we identified a series of issues in the current Internet architecture that motivate a fundamental rethinking of how the Internet should operate in order to cope with new and emerging requirements. Several ICN architectures have been proposed to address some of these requirements, such as the need for efficient information delivery and mobility support. We have shown how ICN research has developed in the last decade, with a major bloom of related activities taking place during the last five years.

Even though the ICN related research area is still shaping, we made an effort to provide a unified view of the alternative proposals by defining a set of core ICN functionalities, e.g., naming, name resolution and data routing, caching, mobility and security. We presented seven ICN architectures, explaining their general goals and operation, as well as how they implement each of these functionalities, culminating with a comparative analysis of the various design choices in each of these areas. This led to a discussion of the open issues for ICN architectures, not only in the core functionalities, but also in other areas that, while important, have so far received much less attention from the majority of ICN efforts.

As a final conclusion, we can state that ICN is a promising and fertile research field that has shown its potential for addressing at least some of the current problems of the Internet, but mostly in a qualitative way so far. There is, therefore, an urgent need for further research and quantitative studies for evaluating the benefits and potential performance gains brought by this new architectural paradigm, as well as for additional work in hitherto neglected areas that are however crucial for the applicability and viability of the ICN paradigm.

References

[1] M. Handley, "Why the Internet only just works," *BT Technology J.*, vol. 24, no. 3, pp. 119–129, July 2006.

[2] J. Rexford and C. Dovrolis, "Future Internet architecture: clean-slate versus evolutionary research," *Commun. ACM*, vol. 53, no. 9, pp. 36–40, September 2010.

[3] P. Stuckmann and R. Zimmermann, "European research on future Internet design," *IEEE Wireless Commun.*, vol. 16, no. 5, pp. 14–22, October 2009.

[4] J. Pan, S. Paul, and R. Jain, "A survey of the research on future Internet architectures," *IEEE Commun. Mag.*, vol. 49, no. 7, pp. 26–36, July 2011.

[5] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "Survey on content-oriented networking for efficient content delivery," *IEEE Commun. Mag.*, vol. 49, no. 3, pp. 121–127, March 2011.

[6] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, July 2012.

[7] Cisco. (2011, June) Visual networking index: Forecast and methodology, 2010-2015. White Paper. [Online]. Available: http://www.cisco.com/go/vni

[8] Google. (2008, July) We knew the web was big. [Online]. Available: http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html

[9] M. Gritter and D. R. Cheriton, "An architecture for content routing support in the Internet," in *USENIX Symposium on Internet Technologies and Systems (USITS)*, 2001.

[10] Stanford University TRIAD project. [Online]. Available: http://www-dsg.stanford.edu/triad/

[11] A. Carzaniga and A. L. Wolf, "Content-based networking: A new communication infrastructure," in *NSF Workshop on an Infrastructure for Mobile and Wireless Systems*, 2001, pp. 59–68.

[12] ——, "Forwarding in a content-based network," in *ACM SIGCOMM*, 2003, pp. 163–174.

[13] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, "A routing scheme for content-based networking," in *IEEE INFOCOM*, 2004, pp. 918–928.

[14] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *ACM SIGCOMM*, 2002, pp. 73–86.

[15] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "Oceanstore: an architecture for global-scale persistent storage," *ACM SIGPLAN Notices*, vol. 35, no. 11, pp. 190–201, November 2000.

[16] M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron, "Virtual ring routing: network routing inspired by DHTs," in *ACM SIGCOMM*, 2006, pp. 351–362.

[17] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, and I. Stoica, "ROFL: routing on flat labels," in *ACM SIGCOMM*, 2006, pp. 363–374.

[18] S. Arianfar, P. Nikander, and J. Ott, "On content-centric router design and implications," in *ACM ReARCH*, 2010.

[19] M. Diallo, S. Fdida, V. Sourlas, P. Flegkas, and L. Tassiulas, "Leveraging caching for Internet-scale content-based publish/subscribe networks," in *IEEE ICC*, 2011, pp. 1–5.

[20] European Community Future Internet Architecture (FIArch) Experts Group. (2011, March) Fundamental limitations of current Internet and the path to future Internet. [Online]. Available: http://www.future-internet.eu/publications/view/article/fundamental-limitations-of-current-internet.html

[21] P. Wendell and M. J. Freedman, "Going viral: flash crowds in an open CDN," in *ACM Internet Measurement Conference (IMC)*, 2011, pp. 549–558.

[22] Z. Wang, Z. Qian, Q. Xu, Z. Mao, and M. Zhang, "An untold story of middleboxes in cellular networks," in *ACM SIGCOMM*, 2011, pp. 374–385.

[23] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, "Packet caches on routers: the implications of universal redundant traffic elimination," in *ACM SIGCOMM*, 2008, pp. 219–230.

[24] A. Anand, V. Sekar, and A. Akella, "SmartRE: an architecture for coordinated network-wide redundancy elimination," in *ACM SIGCOMM*, 209, pp. 87–98.

[25] S. Seetharaman and M. Ammar, "Characterizing and mitigating inter-domain policy violations in overlay routes," in *IEEE International Conference on Network Protocols (ICNP)*, 2006, pp. 259–268.

[26] P. T. Eugster, P. A. Felber, R. Guerraoui, and A. Kermarrec, "The many faces of publish/subscribe," *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, June 2003.

[27] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: seeing the forest for the trees," in *ACM Workshop on Hot Topics in Networks (HotNets)*, 2011.

[28] Y. Huang and H. Garcia-Molina, "Publish/subscribe in a mobile environment," *Wireless Networks*, vol. 10, no. 6, pp. 643–652, November 2004.

[29] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *ACM SIGCOMM*, 2007, pp. 181–192.

[30] FP7 PURSUIT project. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/

[31] FP7 PSIRP project. [Online]. Available: http://www.psirp.org/

[32] FP7 SAIL project. [Online]. Available: http://www.sail-project.eu/

[33] FP7 4WARD project. [Online]. Available: http://www.4ward-project.eu/

[34] FP7 COMET project. [Online]. Available: http://www.comet-project.org/

[35] FP7 CONVERGENCE project. [Online]. Available: http://www.ict-convergence.eu/

[36] NSF Named Data Networking project. [Online]. Available: http://www.named-data.net/

[37] Content Centric Networking project. [Online]. Available: http://www.ccnx.org/

[38] NSF Mobility First project. [Online]. Available: http://mobilityfirst.winlab.rutgers.edu/

[39] ANR Connect project. [Online]. Available: http://anr-connect.org/

[40] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in content-oriented architectures," in *ACM Workshop on Information-Centric Networking (ICN)*, 2011.

[41] V. Jacobson, "A new way to look at networking," Google Tech Talk, August 2006.

[42] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *ACM CoNEXT*, 2009.

[43] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice over content-centric networks," in *ACM ReArch Workshop*, 2009.

[44] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks," in *Proc. IFIP-TC6 Networking Conference*, 2012.

[45] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *ACM Workshop on Information-Centric Networking (ICN)*, 2012.

[46] M. Meisel, V. Pappas, and L. Zhang, "Ad hoc networking via named data," in *ACM MobiArch*, 2010.

[47] D. Smetters and V. Jacobson, "Securing network content," PARC, Tech. Rep. TR-2009-01, October 2009.

[48] S. DiBenedetto, P. Gasti, G. Tsudik, and E. Uzun, "ANDaNA: Anonymous named data networking application," in *Network and Distributed System Security Symposium (NDSS)*, 2012.

[49] D. Trossen and G. Parisis, "Designing and realizing an information-centric Internet," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 60–67, July 2012.

[50] J. Rajahalme, M. Särelä, K. Visala, and J. Riihijärvi, "On name-based inter-domain routing," *Computer Networks*, vol. 55, no. 4, pp. 975–986, March 2011.

[51] K. V. Katsaros, N. Fotiou, X. Vasilakos, C. N. Ververidis, C. Tsilopoulos, G. Xylomenos, and G. C. Polyzos, "On inter-domain name resolution for information-centric networks," in *Proc. IFIP-TC6 Networking Conference*, 2012.

[52] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, July 1970.

[53] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: line speed publish/subscribe inter-networking," in *ACM SIGCOMM*, 2009, pp. 195–206.

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

[55] V. Sourlas, P. Flegkas, G. S. Paschos, D. Katsaros, and L. Tassiulas, "Storage planning and replica assignment in content-centric publish/subscribe networks," *Computer Networks*, vol. 55, no. 18, pp. 4021–4032, December 2011.

[56] N. Fotiou, K. Katsaros, G. Polyzos, M.Sarela, D. Trossen, and G. Xylomenos, "Handling mobility in future publish-subscribe information-centric networks," *Telecommunication Systems*, to appear.

[57] D. Lagutin, "Redesigning Internet-the packet level authentication architecture," Licentiate Thesis, Helsinki University of Technology, Finland, 2008.

[58] N. Fotiou, G. Marias, and G. Polyzos, "Fighting spam in publish/subscribe networks using information ranking," in *Euro-NF Conference on Next Generation Internet (NGI)*, 2010.

[59] SAIL Project. (2013, January) SAIL deliverable B.3 (3.3): Final NetInf architecture. [Online]. Available: http://www.sail-project.eu/deliverables/

[60] ——. (2011, July) SAIL deliverable B.1 (3.1): The network of information: Architecture and applications. [Online]. Available: http://www.sail-project.eu/deliverables/

[61] M. D'Ambrosio, C. Dannewitz, H. Karl, and V. Vercellone, "MDHT: a hierarchical name resolution service for information-centric networks," in *ACM Workshop on Information-Centric Networking (ICN)*, 2011.

[62] C. Dannewitz, M. DAmbrosio, and V. Vercellone, "Hierarchical DHT-based name resolution for information-centric networks," *Computer Communications*, vol. 36, no. 7, p. 736749, April 2013.

[63] G. Garcia, A. Beben, F. J. Ramon, A. Maeso, I. Psaras, G. Pavlou, N. Wang, J. Sliwinski, S. Spirou, S. Soursos, and E. Hadjioannou, "COMET: Content mediator architecture for content-aware networks," in *Future Network & Mobile Summit*, 2011.

[64] W. K. Chai, N. Wang, I. Psaras, G. Pavlou, C. Wang, G. C. de Blas, F. Ramon-Salguero, L. Liang, S. Spirou, A. Beben, and E. Hadjioannou, "CURLING: Content-ubiquitous resolution and delivery infrastructure for next-generation services," *IEEE Commun. Mag.*, vol. 49, no. 3, pp. 112–120, March 2011.

[65] COMET Project. (2011, December) COMET deliverable 3.2: Final specification of mechanisms, protocols and algorithms for the content mediation system. [Online]. Available: http://www.comet-project.org/deliverables.html

[66] ——. (2011, December) COMET deliverable 4.2: Final specification of mechanisms, protocols and algorithms for enhanced network platforms. [Online]. Available: http://www.comet-project.org/deliverables.html

[67] A. Detti, N. Blefari-Melazzi, S. Salsano, and M. Pomposini, "CONET: A content centric inter-networking architecture," in *ACM Workshop on Information-Centric Networking (ICN)*, 2011.

[68] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, and N. Blefari-Melazzi, "Transport-layer issues in information centric networks," in *ACM Workshop on Information-Centric Networking (ICN)*, 2012.

[69] A. Baid, T.Vu, and D. Raychaudhuri, "Comparing alternative approaches for networking of named objects in the future Internet," in *IEEE Workshop on Emerging Design Choices in Name-Oriented Networking (NOMEN)*, 2012.

[70] T. Vu, A. Baid, Y. Zhang, T. Nguyen, J. Fukuyama, R. Martin, and D. Raychaudhuri, "DMap: A shared hosting scheme for dynamic identifier to locator mappings in the global Internet," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2012, pp. 698–707.

[71] S. Nelson, G. Bhanage, and D. Raychaudhuri, "GSTAR: Generalized storage-aware routing for MobilityFirst in the future mobile Internet," in *ACM MobiArch*, 2011.

[72] M. Walfish, H. Balakrishnan, and S. Shenker, "Untangling the web from DNS," in *Symposium on Networked Systems Design and Implementation (NSDI)*, 2004, pp. 225–238.

[73] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish, "A layered naming architecture for the Internet," in *ACM SIGCOMM*, 2004, pp. 343–352.

[74] R. Moskowitz and P. Nikander, "Host identity protocol architecture," RFC 4423, May 2006.

[75] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, k. claffy, D. Krioukov, D. Massey,

C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, "Named data networking (NDN) project," PARC, Tech. Rep. NDN-0001, October 2010.
[76] G. Prasanna, G. Krishna, and H. Garcia-Molina, "Canon in G major: designing DHTs with hierarchical structure," in *International Conference on Distributed Computing Systems (ICDCS)*, 2004, pp. 263–272.
[77] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," in *ACM SIGCOMM*, 2010, pp. 75–86.
[78] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, "The (in)completeness of the observed Internet AS-level structure," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 109–122, February 2010.
[79] M. Yannuzzi, X. Masip-Bruin, and O. Bonaventure, "Open issues in interdomain routing: a survey," *IEEE Network*, vol. 19, no. 6, pp. 49–56, November-December 2005.
[80] D. Perino and M. Varvello, "A reality check for content centric networking," in *ACM Workshop on Information-Centric Networking (ICN)*, 2011.
[81] C. Tsilopoulos and G. Xylomenos, "Scaling bloom filter-based multicast via filter switching," in *International Symposium on Computers and Communications (ISCC)*, 2013.
[82] W. Yang, D. Trossen, and J. Tapolcai, "Scalable forwarding for information-centric networks," in *International Conference on Communications (ICC)*, 2013.
[83] P. Jokela, H. Mahkonen, C. E. Rothenberg, and J. Ott, "(deployable) reduction of multicast state with in-packet Bloom filters," in *Proc. IFIP-TC6 Networking Conference*, 2013.
[84] A. Wolman, M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy, "On the scale and performance of cooperative web proxy caching," in *Symposium on Operating System Principles (SOSP)*, 1999, pp. 16–31.
[85] S. Ihm and V. S. Pai, "Towards understanding modern web traffic," in *ACM Workshop on Information-Centric Networking (ICN)*, 2011.
[86] G. Carofiglio, V. Gehlen, and D. Perino, "Experimental evaluation of memory management in content-centric networking," in *International Conference on Communications (ICC)*, 2011.
[87] A. Helmy, "A multicast-based protocol for IP mobility support," in *Networked Group Communication (NGC)*, 2000, pp. 29–58.
[88] N. Fotiou, G. Marias, and G. Polyzos, "Towards a secure rendezvous network for future publish/subscribe architectures," in *Future Internet Conference (FIS)*, 2010, pp. 49–56.
[89] ——, "Access control enforcement delegation for information-centric networking architectures," in *ACM Workshop on Information-Centric Networking (ICN)*, 2012.
[90] D. Trossen and A. Kostopoulos, "Exploring the tussle space for information-centric networking," in *Research Conference on Communication, Information and Internet Policy*, 2012.
[91] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *International Teletraffic Congress (ITC)*, 2011, pp. 111–118.
[92] C. Stais, A. Voulimeneas, and G. Xylomenos, "Towards an error control scheme for a publish/subscribe network," in *International Conference on Communications (ICC)*, 2013.
[93] Y. Thomas, C. Tsilopoulos, G. Xylomenos, and G. C. Polyzos, "Multisource and multipath file transfers through publish-subscribe internetworking," in *ACM Workshop on Information-Centric Networking (ICN)*, 2013.
[94] C. Tsilopoulos, I. Gasparis, G. Xylomenos, and G. C. Polyzos, "Efficient real-time information delivery in future Internet publish-subscribe networks," in *International Conference on Computing, Networking and Communications (ICNC)*, 2013.
[95] J. Tateson, T. Burbridge, D. Trossen, and M. Ain. PSIRP deliverable 4.6: Final evaluation report on deployment incentives and business models. [Online]. Available: http://www.psirp.org/publications.html

George Xylomenos is an Assistant Professor of Computer Science at the Athens University of Economics and Business (AUEB) and a member of the Mobile Multimedia Laboratory. He received his B.Sc. in Informatics (1993) from AUEB, and M.S. (1996) and Ph.D. (1999) degrees in Computer Science from the University of California, San Diego (UCSD). His current research interests include information-centric network architectures and protocols, multicast-based and peer-to-peer content distribution, the provision of Quality of Service over wireless and mobile networks and real-time transport protocols for multimedia. He has served on the program committee of various international conferences and workshops, including the ACM SIGCOMM Information-Centric Networking Workshop, and has participated in many EU funded FP6 and FP7 projects, including the pioneering PSIRP and PURSUIT projects which developed a clean-slate Future Internet architecture based on pub/sub.

Christopher N. Ververidis is an experienced telecommunications and wireless networks researcher. He received his B.Sc., M.Sc. and Ph.D. degrees in 2000, 2001 and 2008, respectively, from the Department of Informatics at AUEB. His Ph.D. thesis was on cross-layer energy efficient service discovery protocols for Mobile Ad Hoc Networks and revenue optimization mechanisms for charged service provision in ad hoc environments. From 2009 to 2010 he was a post-doctoral researcher at the Department of Wireless Networks at RWTH Aachen University, Germany, working in network economics and quality of experience for cognitive wireless networks. Since 2011 he is a Senior Researcher at the Mobile Multimedia Laboratory. His current research interests lie in the areas of Resource Management and QoS Support for future network architectures. In 2002 he received the Ericsson Award of Excellence in Telecommunications for his master's thesis on Location Based Services. He has worked in many EU and national projects; he has published 25 refereed articles in journals, books and conferences, and he has been awarded one national patent.

Vasilios A. Siris received a degree in physics from the National and Kapodistrian University of Athens, Greece, in 1990, the M.S. degree in computer science from Northeastern University, Boston, in 1992, and the Ph.D. degree in computer science from the University of Crete, Greece, in 1998. He is an assistant professor in the Department of Informatics, AUEB, since 2009, and a research associate at the Institute of Computer Science of FORTH, since 2002. From 2002 to 2008, he was an assistant professor at the University of Crete. In Spring 2001, he was a visiting researcher at the Statistical Laboratory of the University of Cambridge, and in Summer 2001 and 2006, he was a research fellow at the research laboratories of BT in the UK. His current research interests include resource management and traffic control in wired and wireless networks, traffic measurement and analysis for monitoring quality of service and intrusion/anomaly detection, and architecture of mobile communication systems and future networks. He has served as the general chair or technical program chair for various international conferences and workshops, such as Wired/Wireless Internet Communications 2008, IEEE WoWMoM 2009, HotMESH 2011, and IEEE Broadband Wireless Access 2011. He is currently on the editorial board of the Computer Communications Journal. He is/was the principal investigator and coordinator for many research and development projects funded by the EU, the Greek government, and industry.

Nikos Fotiou is a Ph.D. candidate in Computer Science at AUEB and a member of the Mobile Multimedia Laboratory. He received his Diploma in Information and Communication Systems Engineering 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 participated in the PSIRP and PURSUIT projects as well as in the Euro-NF Network of Excellence. His current research interests include availability of name resolution services, privacy preserving information lookup systems for information centric architectures, and access control mechanisms for distributed content storage.

Christos Tsilopoulos is a Ph.D. candidate at AUEB and a member of the Mobile Multimedia Laboratory. He received his B.Sc. in Informatics from AUEB (2006) and his M.Sc. in Communication Systems and Networks from the National and Kapodistrian University of Athens (2009). In his Ph.D. studies he is conducting research in the area of information-centric network architectures and protocols, focusing on multicast routing and forwarding. His research interests also include distributed systems and software engineering. He has participated in several EU funded FP7 projects, including the PSIRP and PURSUIT projects on clean-slate information-centric network architectures.

Xenofon Vasilakos is a Ph.D. candidate at AUEB and a member of the Mobile Multimedia Laboratory. He obtained his B.Sc. in Informatics from AUEB in 2007 and his M.Sc. degree in Parallel and Distributed Systems in 2009 from the Vrije Universiteit, Amsterdam. His current research interests include Future Internet architectures and protocols and especially information-centric networking, with an emphasis on seamless mobility support. He has participated in the EU funded FP7 projects PSIRP and PURSUIT on clean-slate information-centric network architectures.

Konstantinos V. 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 , Professor of Computer Science at AUEB since 1999, has founded and is leading the Mobile Multimedia Laboratory. Previously, 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. While in the US, he participated in large multi-investigator research efforts, such as "Sequoia 2000," and has led many other research projects with funding from industry and the US NSF. He returned at UCSD as Visiting Professor for the 2012-2013 academic year. In Europe he was recently an organizer of the EIFFEL Think Tank, on the Steering Board of the Euro-NF Network of Excellence and head of its "Socio-Economic Aspects" and "Trust, Privacy and Security" Joint Research Activities. He led his lab's participation in EU FP7 projects PURSUIT and PSIRP that developed a clean-slate Future Internet architecture based on pub/sub and the ESA funded projects ϕSAT ("The Role of Satellite in Future Internet Services") and "Service Delivery over Integrated Satellite and Terrestrial Networks." He received his Diploma in Electrical Engineering from the National Technical University of Athens and his M.A.Sc. in Electrical Engineering and Ph.D. in Computer Science from the University of Toronto. His current research interests include Internet architecture and protocols, ubiquitous computing, network security, wireless networks, mobile multimedia communications, and performance evaluation of computer and communications systems. He has been a reviewer for many research funding agencies, on the editorial board and as a guest editor for scientific journals, on the program committees of many conferences and workshops and is currently on the Steering Committee of the ACM SIGCOMM Information-Centric Networking workshop and TPC Co-Chair for ACM SIGCOMM ICN 2013.