A Survey of Information-Centric Networking

Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Börje Ohlman,

ABSTRACT

The information-centric networking (ICN) concept is a significant common approach of several future Internet research activities. The approach leverages in-network caching, multiparty communication through replication, and interaction models decoupling senders and receivers. The goal is to provide a network infrastructure service that is better suited to today's use (in particular. content distribution and mobility) and more resilient to disruptions and failures. The ICN approach is being explored by a number of research projects. We compare and discuss design choices and features of proposed ICN architectures, focusing on the following main components: named data objects, naming and security, API, routing and transport, and caching. We also discuss the advantages of the ICN approach in general.

INTRODUCTION

The increasing demand for highly scalable and efficient distribution of content has motivated the development of future Internet architectures based on named data objects (NDOs), for example, web pages, videos, documents, or other pieces of information. The approach of these architectures is commonly called information-centric networking (ICN). In contrast, current networks are host-centric where communication is based on named hosts, for example, web servers, PCs, laptops, mobile handsets, and other devices.

The ICN architectures leverage in-network storage for caching, multiparty communication through replication, and interaction models that decouple senders and receivers. The common goal is to achieve efficient and reliable distribution of content by providing a general platform for communication services that are today only available in dedicated systems such as peer-to-peer (P2P) overlays and proprietary content distribution networks.

Although the ICN approach was pioneered in TRIAD (www-dsg.stanford.edu/triad/), we base this survey on the following more recent projects representing four approaches being actively developed:

- Data-Oriented Network Architecture (DONA) [1]
- Content-Centric Networking (CCN) [2], currently in the Named Data Networking (NDN) project (www.named-data.org)
- Publish-Subscribe Internet Routing Paradigm (PSIRP) [3], now in the Publish-Subscribe Internet Technology (PURSUIT) project (www.fp7-pursuit.eu)
- Network of Information (NetInf) from the Design for the Future Internet (4WARD) project [4], currently in the Scalable and Adaptive Internet Solutions (SAIL) project (www.sail-project.eu)

While these projects' approaches to ICN differ with respect to their details, they share many assumptions, objectives, and architectural properties. The aim is to develop a network architecture that is better suited for efficiently accessing and distributing content — the currently prevailing usage of communication networks — and that better cope with disconnections, disruptions, and flash crowd effects in the communication service. Communication is driven by receivers requesting NDOs. Senders make NDOs available to receivers by publishing the objects.

As illustrated in Fig. 1, the network can satisfy client requests with data from any source holding a copy of the object, enabling efficient and application-independent caching as part of the network service. The integrity of the delivered data is established independent of the delivering host, which thus can be untrusted.

In this article, we compare and discuss design choices and features of the above four ICN approaches. In CCN, the term *content-centric* is used instead of information-centric, and DONA uses the term *data-oriented*. Henceforth, we use *information*, *content*, and *data* interchangeably.

It should be noted that this article only provides a rough overview comparing the basics of the different approaches. It contains simplifications and may in many cases not take the latest developments of respective approaches into account. This is a consequence of the fact that they are all fairly complex architectures still under development; we are shooting at moving targets.

In the next section, we present the main ICN components and discuss the expected advantages that have motivated ICN. We give a compact overview of the different approaches, followed by a more detailed discussion based on the major ICN components. We conclude with a discussion of remaining challenges.

0163-6804/12/$25.00 © 2012 IEEE

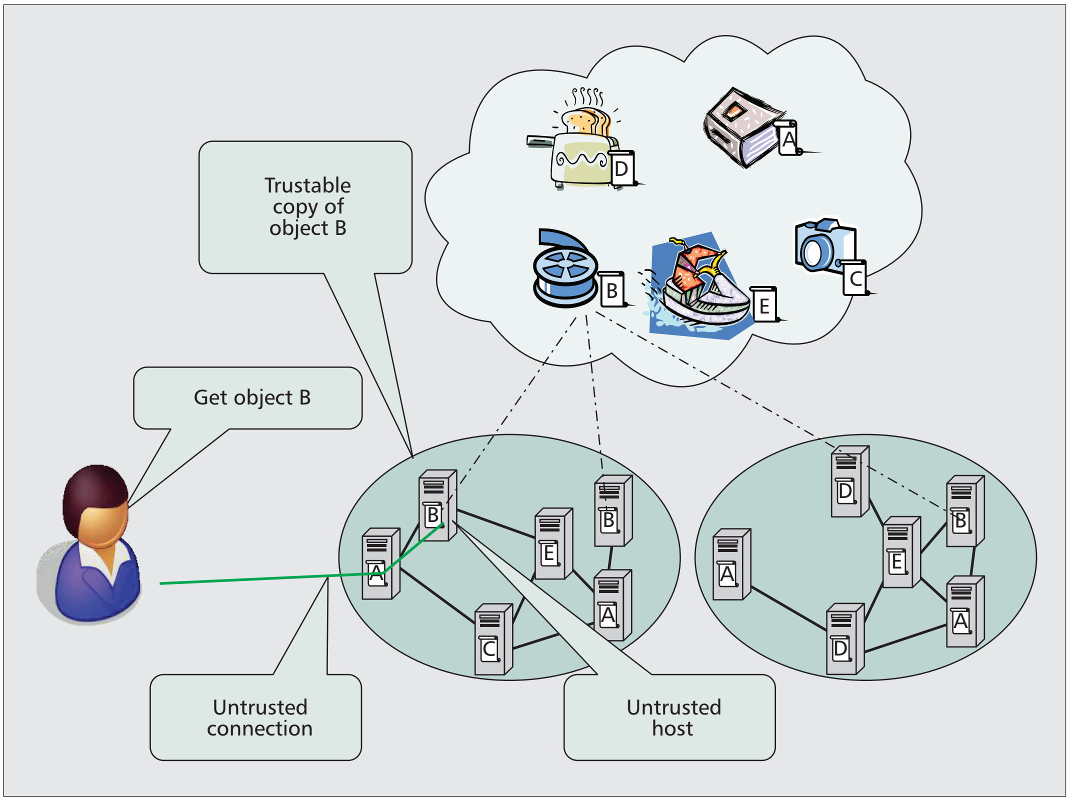

Figure 1. *ICN communication model: client side.*

From a networking perspective, these objects can be viewed as named data chunks without semantics, but some ICN designs have an information abstraction model including multiple representations, i.e., unique bit patterns, for the same object.

The Information-Centric Networking Approach

This section introduces the information-centric approach to networking from a generic perspective by describing the main components common to most proposed designs and their respective design choices, and explaining the advantages motivating the ICN approach.

Main Components of ICN Designs

Named Data Objects — The main abstraction of ICN is the NDO. Examples are web pages, documents, movies, photos, songs, as well as streaming and interactive media; in other words, all types of objects that we store in and access via computers. The NDO is independent of location, storage method, application program, and transportation method. This means that an NDO keeps its name, and thus its identity, regardless of its location and regardless of how it is copied, stored, and communicated. It also means that any two copies of an NDO are for all purposes equivalent; for instance, any node holding a copy can supply it to a requester. *NDO granularity* varies between approaches from packet size to full objects.

From a networking perspective, these objects can be viewed as named data chunks without semantics, but some ICN designs have an *information abstraction model* including multiple representations (i.e., unique bit patterns) for the same object. Examples are different media encodings or different recordings of a piece of music. There can be *meta data* associated with NDOs, such as author, creation date, or other data about the represented information.

Naming and Security — Naming data objects is as important for ICN as naming hosts is for today's Internet. Fundamentally, ICN requires unique names for individual NDOs, since names are used for identifying objects independent of its location or container. It is important to establish a verifiable binding between the object and its name (*name-data integrity*) so that a receiver can be sure received bits actually represent the named object (*object authenticity*). Information about an object's *provenance* (i.e., who generated or published it) is also useful to associate with the name.

The above functions are fundamentally required for the information-centric network to work reliably — otherwise, neither network elements nor receivers can trust objects' authenticity, which would enable several attacks including critical denial of service (DoS) attacks by injecting spoofed content into the network. There are different ways to use names and cryptography to achieve the desired functions [5], and there are different ways to manage namespaces correspondingly.

Two naming schemes have largely been proposed: one with a hierarchical and one with a flat *namespace*. The hierarchical scheme has a structure similar to current URLs, where the hierarchy is rooted in a publisher prefix. The hierarchy enables aggregation of routing information, improving scalability of the routing system. In some cases, the names are *human-readable*, which makes it possible for users to manually type in names, and, to some extent, assess the relation between a name and what the user wants.

The other naming scheme is *self-certifying*, There are delicate design trade-offs for ICN naming affecting routing and security. Self-certifying names are not human readable nor hierarchical. They can, however, provide some structure for aggregation, for instance, a name part corresponding to a publisher.

meaning that the object's name-data integrity can be verified without needing a public key infrastructure (PKI) or other third party to first establish trust in the key. Self-certification is achieved by binding the hash of the content closely to the object's name. This can be done by directly embedding the hash of the content in the name. Another option is an indirect binding, which embeds the public key of the publisher in the name and signs the hash of the content with the corresponding secret key. The resulting names are typically non-hierarchical, or flat, although the publisher field provides structure that can be used for *routing aggregation*.

There are delicate design trade-offs for ICN naming that affect routing and security. Self-certifying names are not human readable or hierarchical. They can, however, provide some structure for aggregation, for instance, a name part corresponding to a publisher. Without self-certification, as mentioned above, the infrastructure depends on a PKI for its operation, which many consider to be a major disadvantage.

Application Programming Interface — The ICN application programming interface (API) is defined in terms of requesting and delivering NDOs. The source/producer makes an NDO available to others by publishing it to the network (called publish or register by the different approaches). A client/consumer asks for an NDO by name (called get, interest, request, find, or subscribe). The latter operation is in most ICN designs a synchronous one-time operation. However, some approaches like PSIRP build on a more publish/subscribe-like approach, where the client registers a subscription and gets notified when something is available.

Both operations (publish and get) use the object's name as the main parameter. In addition, some approaches support supplemental parameters. For example, the CURLING [6] approach supports location preferences for scoping and filtering publications and requests.

Routing and Forwarding — There are two general approaches in ICNs to handle routing, both strongly depending on the properties of the object namespace, in particular, if the names are aggregatable or not. We furthermore distinguish between two routing phases:

- *Routing of NDO requests*
- *Routing of NDO* back to the requester

The first approach uses a name resolution service (NRS) that stores bindings from object names to topology-based locators pointing to corresponding storage locations in the network. This approach has three conceptual routing phases:

- Routing the request message to the responsible NRS node where the object name is translated into one or multiple source addresses
- Routing the request message to the source address(es)
- Routing the data from the source(s) to the requester

All phases can potentially use different routing algorithms. A *name-based routing* method might be used, especially for the first phase. The second and third phases might use topology-based routing like Internet Protocol (IP). There are multiple alternatives to loosely or tightly integrate the phases in an ICN architecture.

The second general approach directly routes the request message from the requester to one or multiple data sources in the network based on the requested object name. The routing algorithm used for this approach heavily depends on the properties of the namespace. After the source has received the request message, the data is routed back to the requester, equaling phase 2 in the NRS-based approach.

Caching — Storage for caching NDOs is an integral part of the ICN service. All nodes potentially have caches, including nodes in operator-run infrastructure networks and user-run home networks, as well as mobile terminals. Requests for NDOs can be satisfied by any node holding a copy in its cache. ICN thus combines caching at the network edge, as in P2P and other overlay networks, with in-network caching (e.g., transparent web caches). The caching is generic, that is, it is application-independent and applies to all providers of content, including user-generated content.

Advantages with the ICN Approach

Efficient content distribution is one advantage of the ICN approach, but as recently argued [7], this alone is not enough to motivate a switch to a new infrastructure. In this section we also describe other advantages that reinforce the motivation.

Scalable and Cost-Efficient Content Distribution — According to recent predictions, global IP traffic will increase by a factor of four from 2010 to 2015, approaching 80 exabytes/mo in 2015. Specifically, global mobile data traffic is expected to increase 26 times between 2010 and 2015. This is mainly attributed to various forms of video (TV, video on demand [VoD], Internet video, and P2P) that will continue to be approximately 90 percent of global consumer traffic by 2015.

The increasing demand for mass distribution and replication of large amounts of resources has led to two main developments: P2P networking and content distribution networks (CDNs). Both approaches represent a move toward a more content-based communication model: uniform resource identifiers (URIs) and DNS names are interpreted in a way that allows accessing cached copies of content in the network. Still, there are a number of issues: suboptimal P2P peer selection that leads to expensive inter-provider traffic, and the inability to effectively leverage in-network storage to reduce overhead for both P2P and CDN scenarios.

Looking at the need for scalable and efficient content distribution, the question is: if users and user agents are more interested in accessing named content, regardless of endpoint locators, is there a more architecturally sound way of addressing these requirements that does not require individual amendments for specific domains and architectures? ICN is the attempt to answer that question "yes."

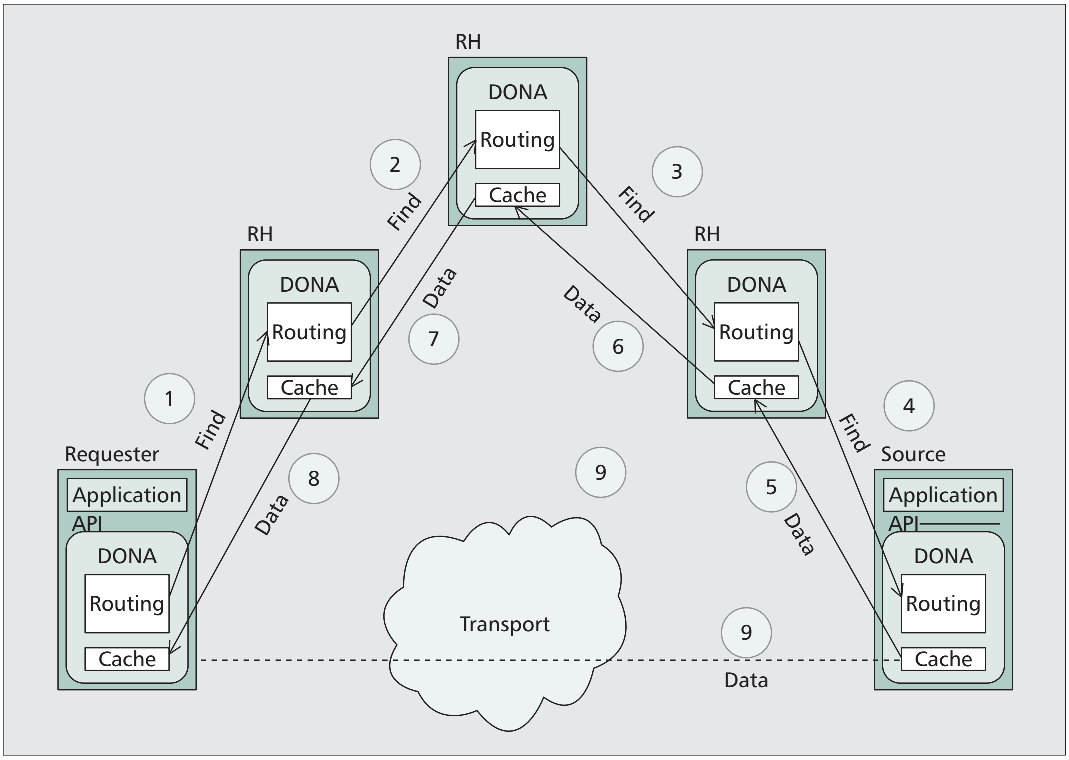

Figure 2. *DONA overview when caching on all resolution handlers (RHs).*

If the primary objective is access to data objects, ICN with its in-network caching can offer store-and-forward approaches similar to the DTN architecture with its convergence layer concept for hop-by-hop transport.

Persistent and Unique Naming — Most content URIs in today's network are actually object locators which, after DNS resolution, exhibit the IP address of a web server that is serving requests by resolving the local part of URI. As a result, the name-object binding can easily break, for example, when an object is moved, the site changes domain, or the site for some reason is unreachable. Moreover, if replicas of the same object are placed at different web servers, they will be accessible using different URIs, and essentially appear as different objects to the system (including caches).

The ICN approach overcomes these problems with persistent and unique naming of NDOs, and with its service model that decouples producers from consumers.

Security Model — Current network security protects the communication channel between a client and a server using Transport Layer Security (TLS) or a similar technique. This security model requires the client to trust the server to deliver correct information over the channel. The ICN security model, in contrast, provides name-data integrity and origin verification of NDOs, independent of the immediate source. The model enables ubiquitous caching with retained name-data integrity and authenticity, something the current model does not provide.

Mobility and Multihoming — The host-based nature of current networks means that mobility and multihoming of nodes and networks become a problem of managing end-to-end connections (e.g., with handovers) and choosing which path or interface to use for these connections.

The ICN approach does not have end-to-end connections that require this kind of connection management. The problem thus becomes much simpler. A moving client just continues to issue requests for NDOs on a new access. Requests on the new access are potentially served from a different source, instead of needing to maintain a connection to the previous source. A multihomed client can similarly choose to send a request on any one, several, or all accesses.

Disruption Tolerance — End-to-end communication with transport sessions to origin servers is often difficult to achieve in challenged networks, with sparse connectivity, high-speed mobility, and disruptions. When application protocol sessions are bound to transport sessions, they will fail as soon as the transport session fails.

Many applications do not require seamless communication with end-to-end paths [8]. If the primary objective is access to data objects, ICN with its in-network caching can offer store-and-forward approaches similar to the data transport networking (DTN) architecture [9] with its convergence layer concept for hop-by-hop transport. This can provide better reliability and better performance by leveraging optimized hop-by-hop transport and in-network caching.

Overview of Information-Centric Network Architectures

This section introduces and illustrates the four ICN approaches at a high level with the purpose of providing a general understanding of them before going into a detailed discussion in the next section.

In CCN, NDOs are published at nodes, and routing protocols are employed to distribute information about NDO location. Routing in CCN can leverage aggregation through a hierarchical naming scheme. NDO security is achieved through Public Key cryptography.

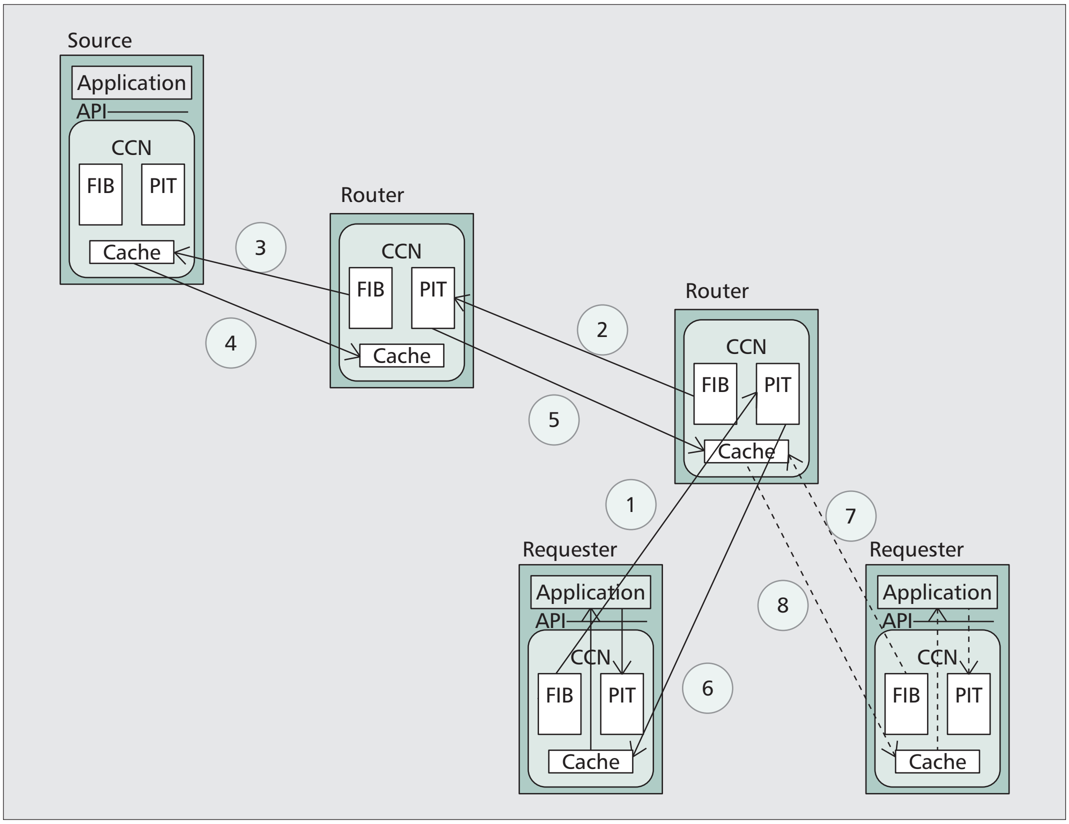

Figure 3. *CCN overview.*

DATA-ORIENTED NETWORK ARCHITECTURE

In DONA, NDOs are published into the network by the sources. Nodes that are authorized to serve data, register to the resolution infrastructure consisting of resolution handlers (RHs). Requests (*FIND* packets) are routed by name toward the appropriate RH, as illustrated in Fig. 2, steps 1–4. Data is sent back in response, either through the reverse RH path (steps 5–8), enabling caching, or over a more direct route (step 9). Content providers can perform a wildcard registration of their principal in the RH, so that queries can be directed to them without needing to register specific objects. It is also possible to register NDO names before the NDO content is created and made available. Register commands have expiry times. When the expiry time is reached, the registration needs to be renewed. The RH resolution infrastructure routes requests by name in a hierarchical fashion and tries to find a copy of the content closest to the client. DONA's anycast name resolution process allows clean support for network-imposed middleboxes (e.g., firewalls, proxies).

CONTENT-CENTRIC NETWORKING

In CCN, NDOs are published at nodes, and routing protocols are employed to distribute information about NDO location. Routing in CCN can leverage aggregation through a hierarchical naming scheme. NDO security is achieved through public key cryptography. Trust in keys can be established via different means, such as a PKI-like certificate chain based on the naming hierarchy, or information provided by a friend. Requests (interest packets) for an NDO are forwarded toward a publisher location, as illustrated in Fig. 3, steps 1–3. A CCN router maintains a pending interest table (PIT) for outstanding forwarded requests, which enables request aggregation; that is, a CCN router would normally not forward a second request for a specific NDO when it has recently sent a request for that particular NDO. The PIT maintains state for all interests and maps them to network interface where corresponding requests have been received from. Data is then routed back on the reverse request path using this state (steps 4–6). CCN supports *on-path caching*: NDOs a CCN router receives (in responses to requests) can be cached so that subsequent received requests for the same object can be answered from that cache (as depicted in steps 7–8, Fig. 3). From a CCN node's perspective, there is balance of requests and responses; that is, every single sent request is answered by one response (or no response). CCN nodes can employ different *strategies* for requests (re-) transmission pace and interface selection depending on local configuration, observed network performance, and other factors. The NDN project advances the CCN approach. It provides a topology-independent naming scheme and is exploring greedy routing for better router routing scalability.

PUBLISH-SUBSCRIBE INTERNET ROUTING PARADIGM

In PSIRP, NDOs are also published into the network by the NDO sources as illustrated in step 1, Fig. 4. The publication belongs to a par-

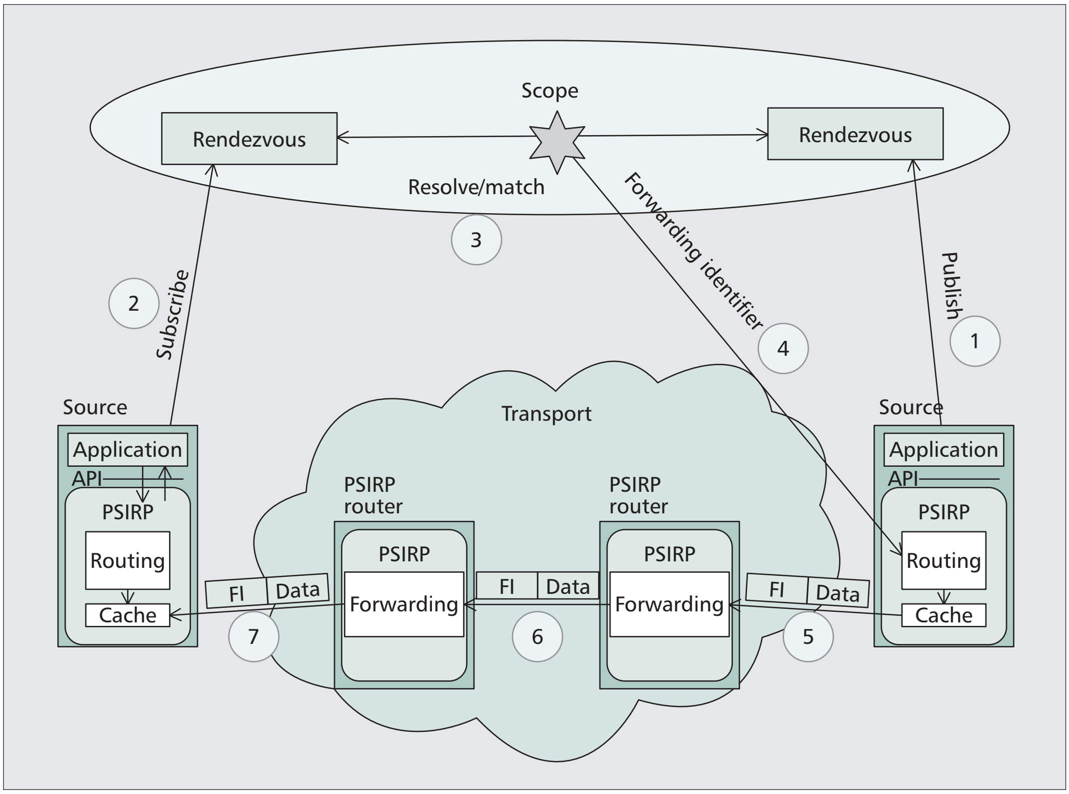

Figure 4. *PSIRP overview*

As soon as a copy is reached, the data will be returned to the receiver. The two models are merged in a hybrid resolution/routing approach where a global resolution system provides mappings in the form of routing hints that enable aggregation of routing information.

ticular named scope. Receivers can subscribe to NDOs (step 2). The publications and subscriptions are matched by a *rendezvous system* (step 3). The subscription request specifies the scope identifier (SI) and the rendezvous identifier (RI) that together name the desired NDO. The identifiers are input to a matching procedure resulting in a forwarding identifier (FI), which is sent to the NDO source (step 4) so that it can start forwarding data (steps 5–7). The FI consists of a Bloom filter that routers use for selecting the interfaces on which to forward an NDO. This means that routers do not need to keep forwarding state. The use of Bloom filters results in a certain number of false positives; in this case this means forwarding on some interfaces where there are no receivers.

Network of Information (NetInf)

NetInf offers two models for retrieving NDOs, via *name resolution* and via *name-based routing*, thereby allowing adaptation to different network environments. In NetInf, depending on the model used in the local network, sources publish NDOs by registering a name/locator binding with a name resolution service (NRS), or announcing routing information in a routing protocol. A NetInf node holding a copy of an NDO (including in-network caches and user terminals) can optionally register its copy with an NRS, thereby adding a new name/locator binding. If an NRS is available, a receiver can first *resolve* an NDO name into a set of available locators and can subsequently retrieve a copy of the data from the “best” available source(s), as illustrated in steps 1–4 of Fig. 5. Alternatively, the receiver can directly send out a *GET* request with the NDO name, which will be forwarded toward an available NDO copy using name-based routing (steps 5–8 in the figure). As soon as a copy is reached, the data will be returned to the receiver. The two models are merged in a hybrid resolution/routing approach where a global resolution system provides mappings in the form of routing hints that enable aggregation of routing information.

Design Choices and Trade-Offs

This section provides an in-depth discussion of the different ICN approaches based on the following architectural aspects: naming and security, API, name resolution and routing, caching, transport, and mobility.

Naming and Security for Data Objects

DONA names NDOs with a flat namespace in the form *P:L*, where *P* is the globally unique *principal* field, which contains the cryptographic hash of the publisher’s public key, and *L* is the unique object label. As *P* identifies the *publisher* (and not the owner), republishing the same content by a different publisher (e.g., by an in-network cache) generally results in a different name for the same content. While this can be circumvented with specific means in DONA (e.g., via wildcard queries or principal delegation), it might complicate benefiting from all available content copies.

The CCN namespace is hierarchical in order to achieve better routing scalability through name-prefix aggregation. The names are rooted in a prefix unique to each publisher. The publisher prefix makes it possible for clients to con-

CCN names typically do not contain the publisher's PK (nor its cryptographic hash). The hash of static content is typically also not explicitly part of the name used by requesters. While this improves human readability, it complicates self-certification.

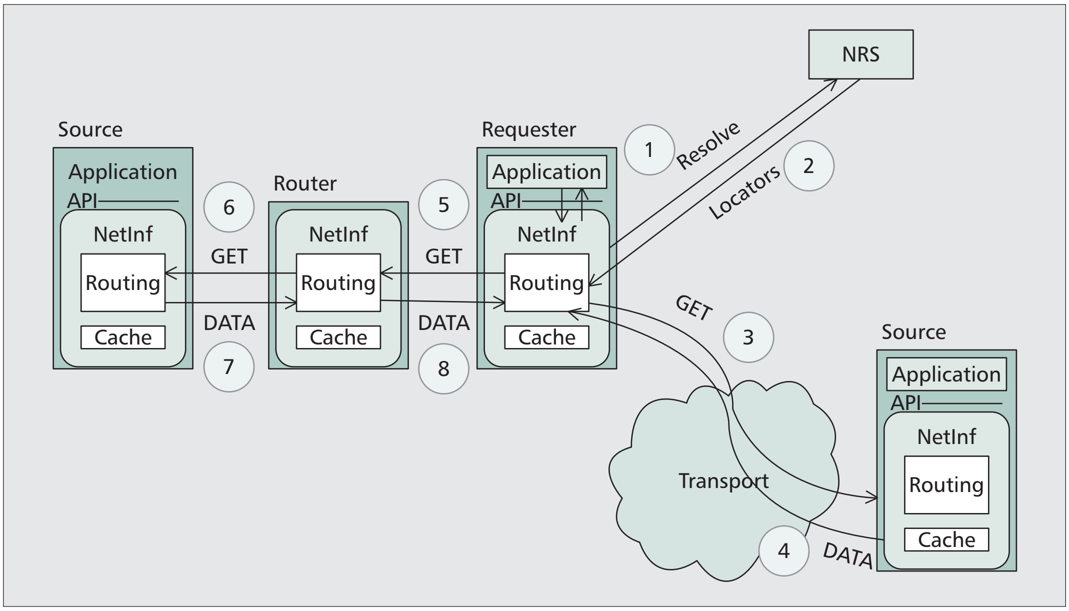

Figure 5. *NetInf overview.*

struct valid names for data that does not yet exist, and publishers can respond with dynamically generated data. CCN names are used for both naming information and routing purposes. The granularity of the names is very fine: single chunks (packets) are named.

PSIRP makes use of two types of names: *rendezvous identifiers* and *scope identifiers*; they both belong to a flat namespace. Rendezvous identifiers (together with scope identifiers) name NDOs. The NDOs are mapped to rendezvous points, which are used to establish contact between publishers and subscribers. PSIRP also uses *forwarding identifiers*, which are used by the forwarding fabric to transport data after contact is established at a rendezvous point. The forwarding identifiers (Bloom filters in LIPSIN [10]) are not names for NDOs; they are transient and identify a path from the publisher to the subscriber.

NetInf generally employs a flat namespace [11] with some structure similar to the DONA namespace. In order to accommodate different ICN deployment requirements, NetInf distinguishes between a common naming format (that all nodes must understand) and name semantics and name-object-binding validation mechanisms. The common NetInf naming format [12] is based on containing hash digests in the name, and different hashing schemes (e.g., SHA2-based object content digests) are supported. The hash digest of the owner's public key (PK) can also be contained in the name to support dynamic data. NetInf names can be transformed to different representations, including a URI representation and a binary representation.

DONA, NetInf, and PSIRP use flat namespaces. All three can check the name-data integrity solely based on the data's name (i.e., without requiring external means like a PKI). This property is called *self-certifying names*. To achieve this for static data, the cryptographic hash of the content can be included as an object label. For dynamic data, name-data integrity is achieved by providing a signature of the content's hash as metadata with the NDO, signed by the public key corresponding to the hash in the ID's authenticator field. In this way, the object identifier is securely bound to the data and also allows data to be handled that does not yet exist. Using self-certifying identifiers is a deliberate trade-off between desirable name-data integrity properties and human readability as the identifiers contain a cryptographic hash. As a result, additional means are potentially required to securely bind more human-readable application-level names to these identifiers. One could argue that many URLs in today's Internet already suffer from the same problem of human unfriendliness. On the other hand, self-certifying identifiers allow checking whether the received data matches the identifier used in the data request *without* requiring a PKI, which simplifies the security model and makes it more reliable. For example, no trust in the PKI is required, and data integrity can be verified offline.

CCN names typically do not contain the publisher's PK (or its cryptographic hash). The hash of static content is typically also not explicitly part of the name used by requesters. While this improves human readability, it complicates self-certification. Data integrity is also achieved by signing the content with the publisher's secret key, but trust in the signing key always needs to be established using external means since there is no direct binding between the key and the NDO name. CCN supports multiple different means to verify trust in the key, such as direct experience, information provided by friends, a trusted directory of keys, or a global PKI.

Application Programming Interface

The API for ICN is fundamentally different from today's socket API of TCP/IP. The latter is designed for establishing a communication channel between two endpoints at specific locations. An ICN API is designed to allow applications to request an NDO from the network without requiring any knowledge of the object's network location and the retrieval approach (e.g., multisource download).

The main API calls of all presented ICN approaches are the equivalent to a publish and a get call as outlined earlier. However, these calls address different underlying network entities. In PSIRP and NetInf, publications are addressed to the rendezvous system and NRS, respectively, to register new names, resulting in corresponding binding entries. In CCN and DONA, publish is used to fill the routing tables of the content routers/RH. Likewise, in NetInf and PSIRP, the get calls are addressed to the resolution/rendezvous system, followed by a second step to retrieve the data from the NDO source. This second step is, however, typically hidden from the API as locators and routing hints are typically not exposed to ICN applications. In CCN and DONA, the get call is handled directly by the routers/RH. NetInf is a hybrid approach as it can send a publish/get to the NRS for name resolution as well as directly to a router for name-based routing. This is explained in more detail in the next section.

Name Resolution and Routing

In ICN there are two key functions that name resolution and routing must achieve when there is a request for a specific NDO. The first is to find a node that holds a copy of the NDO and deliver the request to that node (i.e., routing of NDO requests). The second is to find a path from that node back to the requester over which the NDO can be delivered (i.e., routing of NDOs). One way to do this is through *name resolution*, which means that a resolution service is queried, and one or more lower-layer locators are returned. These locators can then be used to retrieve the object, using a protocol like HTTP or direct IP. An alternative is to directly forward the request to an object copy in the network based on the object name, without first resolving the object name into some lower-layer locators. This approach is often referred to as *name-based routing*. Note that name resolution might also include some steps that involve name-based routing, such as when a DHT-based name resolution system is used.

DONA uses name-based routing to route the query via the RHs to a copy of the requested NDO. Nodes that are authorized to serve data use the `REGISTER`(P:L) primitive to register a datum with an RH. Each domain/publisher has an RH. To resolve a name, the `FIND`(P:L) primitive is used. Both primitives allow for wildcards being used in place of P or L. RHs are organized in a hierarchical structure. Every request that an RH cannot handle is forwarded to its parent RH. The RH tries to find a copy of the content closest to the client. Once a copy is found, the data is returned to the client, potentially via the RH request path as shown in Fig. 2 when the RH performs caching. Otherwise, the data can also be returned directly to the client. Originally, DONA used longest-prefix matching for name matching, currently the more scalable deepest-match approach is being proposed [5].

CCN uses name-based routing. Clients ask for a data object by sending interest packets, which are routed toward the publisher of the name prefix using longest-prefix matching in the forwarding information base (FIB) of each node. The FIB can be built using routing protocols similar to those used in today's Internet. The CCN nodes keep state for each outstanding request in the pending interest table (PIT; Fig. 3). This makes request aggregation possible, i.e., when the same node receives multiple requests for the same NDO, only the first is forwarded towards the source. When a copy of the data object is encountered on the path, a data packet containing the requested object is sent on the reverse path back to the client (all nodes along the path cache a copy of the object). The reverse path is found using the state that the interest packet has left in the nodes.

PSIRP uses a resolution model where the resolver is called the rendezvous point. The data return path to the client can, potentially, take a different path than the name resolution/rendezvous path. The rendezvous point does not have to be on the path to the publisher or hold a copy of the data. Data is forwarded using a source routing approach called *zFilters*: a Bloom filter describing the route is built by the rendezvous point and used to forward packets from the selected source to the destination. The Bloom filter is attached to the packet itself, and it contains all names of the links that have to be followed. The Bloom filter approach allows packet length to be traded off against wasting network resources. A large Bloom filter gives fewer false positives, thus resulting in less packets being forwarded on links without any receiver.

NetInf represents a hybrid architecture that supports name resolution as well as name-based routing to retrieve data objects. NetInf supports a wide variety of name resolution services to have flexibility and scalability to cater for a wide range of network types. The same mechanism that works in a small ad hoc network (e.g., simple mechanisms like broadcast) might not work/scale in a global core network. NetInf defines an interdomain interface for name resolution and routing that allows using different mechanisms in different parts of the network. Two explicit name resolution mechanisms have been developed so far. The first is called Multilevel Distributed Hash Table (MDHT) [13]. It uses a topologically embedded hierarchy of resolvers, potentially distributed hash tables (DHTs), for enabling scalable and location-aware resolution of flat namespaces. In addition, NetInf provides an NRS approach called Late Locator Construction (LLC) [14] that focuses on handling highly dynamic network topologies, including large moving networks. The name resolution approach of NetInf allows for smooth evolution from today's Internet. The NRS can resolve object names into traditional URL, which can be retrieved using the existing HTTP protocol.

Comparing the alternatives, one can note that a name-resolution-based approach has the advantage that caching is not limited to copies on the forwarding path. Instead, the NRS can manage and provide locators for any available object copy, including nearby (but off-path) in-network caches and copies stored on user devices

In CCN and DONA, the get call is handled directly by the routers/RH. NetInf is a hybrid approach as it can send a publish/get to the NRS for name resolution as well as directly to a router for name-based routing.

	DONA	CCN	PSIRP	NetInf
Namespace	Flat with structure	Hierarchical	Flat with structure	Flat with structure
Name-data integrity	Signature, PKI independent	Signature, external trust source	Signature, PKI independent	Signature or content hash, PKI indep.
Human-readable names	No	Possible	No	No
Information abstraction model	No	No	No	Yes
NDO granularity	Objects	Packets	Objects	Objects
Routing aggregation	Publisher/explicit	Publisher	Scope / explicit	Publisher
Routing of NDO request	Name-based (via RHs)	Name-based	NRS (rendezvous)	Hybrid NRS and name-based
Routing of NDO	Reverse request path or direct IP connection	Reverse request path using router state	Source routing using Bloom filter	Reverse request path or direct IP connection
API	Synchronous get	Synchronous get	Publish/subscribe	Synchronous get
Transport	IP	Many including IP	IP/PSIRP	Many including IP

Table 1. *Summary of characteristics of the ICN approaches.*

that can be accessed, say, in local scenarios. In addition, an NRS-based approach can simplify migration as the routing and forwarding underlay does not have to be modified. On the other hand, name-based routing eliminates the name resolution step completely, thereby potentially reducing the overall latency and simplifying the overall process. It is also not clear how interest aggregation can be done in an efficient way in an NRS-based approach.

Caching

ICN generally leverages in-network storage to provide a better-performing and more robust transport service. NDOs can be cached on-path (e.g., as in traditional web caching), but ICN can also make cached objects available for off-path requests by announcing them in a routing protocol or by registering them in a name-resolution service. In addition to caching NDOs, some ICN approaches do *request aggregation*, which can be seen as a form of request caching.

In DONA, caching is inherent in the architecture. Any RH can also serve as a cache. To populate its cache, the RH modifies the FIND request so that the NDO is returned to the RH before it is returned to the original requester. Any cache can respond to a FIND request by returning a cached copy of the NDO.

CCN can cache both requests (through its request aggregation) and objects. CCN routes a request for data toward the publisher, and makes use of any cached copies along that path. A CCN node can keep received interest packets in a *pending interest table* and thus suppress forwarding of subsequently received requests for the same object if it has already sent an request. Object copies can also be found by local search. As single packets are the atomic objects in CCN, it is possible that only a part of a bigger object is cached.

In PSIRP, caching is limited to the scope of the rendezvous point for the identifier associated with an object. Within that scope an object can be cached in multiple caches.

NetInf can cache requests and objects, too. Generally, name-based routing and name resolution is employed to find next-hop options. When a node receives a GET request, it can decide to employ a pending interest-table-like structure for request aggregation. It can also decide to perform a dedicated NRS lookup for each received interest, so performing request aggregation becomes a policy decision. There are two ways to make use of a cached object copy: first, any copy can be found directly by querying the NRS, provided that the copy is explicitly registered there or discovered by the NRS via other means (e.g., broadcast); second, the copy can be found by a cache-aware NetInf transport protocol on the path to a location known to hold a copy (e.g., a location retrieved from the name resolution system).

Transport

By transport we refer to two concepts:

- The fundamental request and response forwarding mechanisms
- Transport protocol functions such as resource sharing (“congestion control”), flow control, and reliability

The DONA architecture does not put much emphasis on transport and seems to rely on existing transport protocols such as TCP.

CCN defines different packet types, *interest packets* and *data packets*, representing basic elements of a protocol. A node sending an interest packet via one of its interfaces (called *faces* in CCN) to a (set of) neighbor nodes has some expectation to receive a corresponding data packet shortly (i.e., CCN nodes operate on the principle that there is balance of interest and data packets). Interest and data packets work on the packet level — the assumption is that larger objects would be represented by individual chunks, and each chunk can be accessed by a unique name.

This fundamental mechanism is CCN's basis for realizing different services that are conventionally considered "transport layer functions," such as reliable transmission, flow control, and multipath communication. It is essentially up to a node's specific *strategy* to which face interest packets should be sent and how to behave in reaction to the received data packets.

PSIRP's basic forwarding mechanism is based on Bloom filters as described earlier. PSIRP proposes to use different names for each object to handle flow control. These names are derived algorithmically from the original name, encoding the desired receiving speed. Another option is for the receivers to publish flow control feedback under some algorithmically derived name for the sender to possibly subscribe to.

NetInf defines a set of messages to request resources and reply to these requests, for example, by returning the requested object, or returning a locator or redirection hint. This protocol is implemented by convergence layers; that is, concrete wire protocols for specific underlays. There could be multiple hops involved in forwarding such request and response messages, and each hop can potentially use a different convergence layer.

The convergence layer used in this hop-by-hop approach implements (or employs) a specific transport protocol that provides the appropriate resource sharing and reliability mechanisms for the corresponding network path (segment). This approach allows for localized transport mechanisms (i.e., for challenged wireless links) without degrading performance on other hops.

MOBILITY

In this section, we discuss three types of mobility and how they relate to information-centric networks. The first type is *client mobility*: a client moves during or between requesting data objects. The second type is *content mobility*: an object or set of objects changes location. The third type is *network mobility*: when an entire network moves (e.g., a body area network or a train network).

A key feature with information-centric networks is that all copies are equal. For client mobility this means that when a client moves there is no need to keep an association to a specific copy alive. Instead, new associations can be established to alternative copies close to the new location. All discussed ICN approaches can find a new appropriately located copy easily when a client moves.

When the content (the publisher) moves, the routing information in the network needs to be updated. For NRS-based approaches like NetInf, this only means that a new locator is registered when an NDO is published with a new location in the network. For approaches like CCN that use name-based routing and hierarchical naming to aggregate route announcements, the situation is more problematic. If the full aggregate of objects is moving, the new route announcement needs to be propagated and old routing entries need to be replaced before routing converges, causing similar issues that we see in today's IP networks. In the case when only parts of the objects belonging to an aggregate move to a new location (e.g., a company employee takes a laptop on a trip), there will also be a need to fragment the routing tables. In agile network scenarios, this could defeat the benefits of having a hierarchical namespace. In PSIRP and DONA, content mobility involves updating the routing state in the rendezvous nodes and RH, respectively. However, both do not suffer from the aggregation problem thanks to their flat namespaces.

Moving an entire network can cause a storm of routing/resolution updates, especially if the moving network consists of a heterogeneous set of publishers, as in a train. This can be problematic for both name-based routing schemes as well as NRS-based approaches if they do not allow for relative route announcement or relative locators. For example, if the publishers can express their location as relative to the location of the moving network, only the location of the network needs to be updated when it moves, not all the locations of objects currently attached to the moving network. An example of such a routing/forwarding system supporting relative locators is the NetInf LLC resolution system [14].

NetInf can support all three types of mobility. Content/content provider mobility is supported via the NRS. When a data copy moves, this movement results in an update in the NRS to account for the new network location. NRS updates are a standard operation in NetInf, which can be performed fast and do not result in inflated lookup tables. The handling of client mobility heavily depends on the data transport and forwarding technology used in NetInf. In general, NetInf can support different data transport and forwarding technologies. For example, the integrated NRS and routing/forwarding system Global Information Network (GIN) [15] natively supports client mobility without inflating the routing tables. The alternative NetInf routing/forwarding system LLC provides very good support for network mobility.

In PSIRP, clients can just unsubscribe, switch networks, and resubscribe again. A new path/subtree will be computed by the routing layer. Buffering and sequence numbering allow for seamless handovers. Content provider mobility is more complex and involves updating the routing state in the rendezvous nodes.

Client mobility in CCN is inherent. A client can switch to another network and continue to issue interest packets. The strategy layer could notice the switch and re-issue all the pending interests, without waiting for them to timeout. Content provider mobility is more complex: a content provider would have to update the routing tables of all relevant neighboring nodes. Furthermore, moving content providers would pollute the routing tables with specific prefixes, countering the advantage of prefix aggregation.

Client mobility in DONA is achieved in a way similar to PSIRP: clients can deregister from their previous location and reregister at the new location. Deregistration is not mandatory, as res-

NetInf defines a set of messages to request resources and to reply to these requests; for example, by returning the requested object or by returning a locator or redirection hint. This protocol is implemented by convergence layers, i.e., concrete wire protocols for specific underlays.

olution handlers can expunge outdated content entries.

The incentives for all involved players have to be clearly communicated to foster deployment. It can be supported by standardizing central ICN aspects. Deployment is also simplified if the ICN architecture allows to grow from the edges, i.e., incremental deployment is supported.

Conclusion

We have explained and motivated the ICN approach to the network of the future. Based on the designs proposed in the research community, we have discussed the major trade-offs. Table 1 summarizes the different properties and design choices of the analyzed approaches according to the previous discussion.

There are, however, remaining challenges that need to be addressed for the ICN ideas to become deployed and used on a wider scale.

Scalability: The number of NDOs is vastly larger than the number of hosts in the current Internet, which means the ICN routing and name resolution system has a harder job than today's global IP routing and DNS name resolution. It remains to be shown that the proposed means for aggregation of routing information and scalability of name resolution works in practice.

Privacy: Requests for content are visible to the ICN network, resulting in a possibly worse privacy situation than exists today. On the other hand, it might not be possible to relate a request to a particular person. The privacy issues need to be investigated in more detail in order to understand the full consequences and find means to mitigate them.

Legal issues: Ubiquitous caching probably does not sound too appealing for some content owners, who fear that their content can be illegally spread. Can the combination of technical mechanisms and new laws and regulation provide an acceptable solution?

Deployment: The incentives for all involved players have to be clearly communicated to foster deployment. It can be supported by standardizing central ICN aspects (e.g., naming [12]). Deployment is also simplified if the ICN architecture allows growth from the edges (i.e., incremental deployment is supported).

Acknowledgments

The authors would like to thank the partners from the 4WARD and SAIL projects for their contributions in many discussions on ICN, the participants of the 2010 Dagstuhl seminar on ICN (http://www.dagstuhl.de/10492) for their feedback on our initial survey, and the anonymous reviewers for their constructive feedback.

References

[1] T. Koponen *et al.*, "A Data-Oriented (and Beyond) Network Architecture," *Proc. SIGCOMM '07*, Kyoto, Japan, Aug. 27–31, 2007.

[2] V. Jacobson *et al.*, "Networking Named Content," *Proc. CoNEXT*, Rome, Italy, 2009, http://doi.acm.org/10.1145/1658939.1658941, pp. 1–12.

[3] M. Ain *et al.*, "D2.3 – Architecture Definition, Component Descriptions, and Requirements," Deliverable, PSIRP 7th FP EU-funded project, Feb. 2009.

[4] B. Ahlgren *et al.*, "Second NetInf Architecture Description," 4WARD EU FP7 Project, Deliverable D-6.2 v2.0, Apr. 2010, FP7-ICT-2007-1-216041- 4WARD / D-6.2, http://www.4ward-project.eu/.

[5] A. Ghodsi *et al.*, "Naming in Content-Oriented Architectures," *Proc. ACM SIGCOMM Wksp. Information-Centric Networking*, Toronto, Canada, Aug. 2011.

[6] W. Chai *et al.*, "Curling: Content-Ubiquitous Resolution and Delivery Infrastructure for Next-Generation Services," *IEEE Commun. Mag.*, vol. 49, no. 3, 2011; http://discovery.ucl.ac.uk/1305428/, pp. 112–20.

[7] A. Ghodsi *et al.*, "Information-Centric Networking: Seeing the Forest for the Trees," *HotNets-X*, Cambridge, MA, Nov. 2011.

[8] J. Ott and D. Kutscher, "Why Seamless? Towards Exploiting WLAN-based Intermittent Connectivity on the Road," *Proc. TNC '04*, Rhodes, June 2004.

[9] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," *Proc. ACM SIGCOMM '03*, Karlsruhe, Germany, Aug. 25–29, 2003, pp. 27–36.

[10] P. Jokela *et al.*, "LIPSIN: Line Speed Publish/Subscribe Inter-Networking," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 39, no. 4, 2009, pp. 195–206.

[11] C. Dannewitz *et al.*, "Secure Naming for A Network of Information," *Proc. 13th IEEE Global Internet Symp. '10*, San Diego, CA, Mar. 2010.

[12] S. Farrell *et al.*, "Naming Things with Hashes," IETF Internet draft draft-farrell-decade-ni, work in progress, Apr. 2012; http://tools.ietf.org/html/draft-farrell-decade-ni.

[13] M. D'Ambrosio *et al.*, "MDHT: A Hierarchical Name Resolution Service for Information-Centric Networks," *Proc. ACM SIGCOMM Wksp. Information-Centric Networking*, Toronto, Ontario, Canada: ACM, 2011; http://doi.acm.org/10.1145/2018584.2018587, pp. 7–12.

[14] A. Eriksson and B. Ohlman, "Scalable Object-to-Object Communication Over A Dynamic Global Network," *Proc. Future Network and Mobile-Summit '10*, June 2010.

[15] M. D'Ambrosio *et al.*, "Providing Data Dissemination Services in the Future Internet," *Proc. WTC '08*, New Orleans, LA, Dec. 2008, at IEEE GLOBECOM '08.

Biographies

BENGT AHLGREN leads the Communication Networks and Systems laboratory at SICS and directs the SICS Center for Networked Systems. He received his Ph.D. in computer systems in 1998 from Uppsala University, Sweden. He conducts research in the area of computer networking including the protocols and mechanisms of the Internet infrastructure. His main interest is the evolution of the Internet architecture, especially issues with naming and addressing on a global scale. Lately his research focus is on designing networks based on an information-centric paradigm.

CHRISTIAN DANNEWITZ studied computer science and electrical engineering at the University of Paderborn, Germany, and Carleton University, Canada. He currently drives the research on ubiquitous and information-centric networking at the Computer Networks Group, University of Paderborn, and expects to complete his PhD in 2012. Previously, he led the research of a software company, focusing on information management. His research interests include networking and future Internet technologies with a focus on architecture and information-centric networking.

CLAUDIO IMBRENDA has had an interest in computers since he was very young. He received a Master's degree in computer science, with honours, from the University of Pisa. He joined NEC Laboratories Europe as a software engineer in 2010, working on the SAIL EU-FP7 project, for which he implemented a prototype.

DIRK KUTSCHER is a senior researcher at NEC Laboratories Europe. He received his doctoral degree (Dr.-Ing.) from Universität Bremen in 2003, while working for the Center for Computing Technologies (Technologiezentrum Informatik, TZI). His research interest is focused on technologies than can help to evolve the Internet to better support robust mobile communications and information-centricity. He leads the NetInf work package in the SAIL EU-FP7 project and is co-chairing the IRTF Research Group on Information-Centric Networking.

BÖRJE OHLMAN has an interest in computers that goes back to the late 1970s. He joined Ericsson in the 1980s to develop signaling for ATM, and was active in standard bodies (ITU-T, ETSI, and ATM Forum). In the 1990s he worked with IP, including IETF standardization. He has been working on future networking technologies in EU-sponsored projects Ambient Networks, 4WARD, and SAIL since 2005. In 4WARD he co-lead the NetInf workpackage. He is co-chairing the newly formed IRTF research group information-centric networking (ICNRG).