A Survey of Information-Centric Networking (Draft)

Bengt Ahlgren Christian Dannewitz Claudio Imbrenda
Dirk Kutscher Börje Ohlman (in alphabetical order)

February 2, 2011

1 Introduction

The development of the Information-Centric Networking (ICN) concept is one of the significant results of different international Future Internet research activities. In this concept, the principal paradigm is not end-to-end communication between hosts – as in the current Internet architecture. Instead, an increasing demand for highly scalable and efficient distribution of content has motivated the development of architectures that focus on information objects, their properties, and receiver interest in the network to achieve efficient and reliable distribution of such objects. Corresponding network architectures can leverage in-network storage, multiparty communication through replication and interaction models such as publish-subscribe to provide general platforms for communication services that are today only available in dedicated systems such as peer-to-peer overlays and proprietary content-distribution networks.

The information-centric approach to the network of the future has recently been and is being explored by a number of research projects, both in Europe (PSIRP [1], 4WARD [2, 3], PURSUIT[1] and SAIL[2]) and in the US (CCN [4], DONA [5] and NDN[3]). While these approaches differ with respect to their specific architecture, they share some assumptions, objectives and certain structuring architectural properties. In general, the aim is to develop network architectures that are better suited for content distribution (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* information objects. The network can satisfy the request with data from any source holding a copy of the object, enabling efficient and application-independent caching as part of the network service. Senders make information objects available to receivers by *publishing* the objects. This decoupling in space and time between senders and receivers require that information objects carry security metadata for verifying the objects' integrity and authenticity.

In this paper we compare and discuss some of the features and design choices of the 4WARD Networking of Information architecture (NetInf), PARC's

[1] `http://www.fp7-pursuit.eu/`
[2] `http://www.sail-project.eu/`
[3] `http://www.named-data.net/`

Dagstuhl Seminar Proceedings 10492
Information-Centric Networking
http://drops.dagstuhl.de/opus/volltexte/2011/2941

Content Centric Networking (CCN), the Publish-Subscribe Internet Routing Paradigm (PSIRP), and the Data Oriented Network Architecture (DONA). All four projects take an information-centric approach to designing a future network architecture, where the information objects themselves are the primary focus rather than the network nodes. In CCN the term *content-centric* is used with essentially the same meaning. Henceforth we therefore use *information* and *content* interchangeably.

In the next chapter we present some problems that motivated the work on information-centric networking. In Chapter 3 we discuss the building blocks of a generic ICN architecture, followed in Chapters 4 and 5 by an overview and comparison of the architectures from the four projects mentioned above, covering their major components and properties. In Chapter 6 we discuss incentives for deploying ICN technology. In Chapter 7 we cover remaining challenges for research on ICN.

2 Problem statement

The Information-Centric Networking (ICN) paradigm is addressing a set of issues of the current Internet and content distribution architectures. In general, the need for scalable and efficient content distribution has fueled a proliferation of overlay networks such as Peer-to-Peer (P2P) networks and Content Distribution Networks (CDNs). This has introduced information access models where endpoints essentially try to access named content, without actually requiring a transport layer session that is bound to a specific host (or even a network interface).

Moreover, there are other concerns, e.g., about the scalability of the Internet routing infrastructure [6] that have led to research of alternative addressing and routing approaches that would not be constrained by the current host-locator-based routing.

2.1 Scalable and Cost-Efficient Content Distribution

According to recent predictions[4], global IP traffic will increase by a factor of four from 2009 to 2014, approaching 64 exabytes per month in 2014, compared to approximately 15 exabytes per month in 2009. Specifically, global mobile data traffic is expected to double every year through 2014, increasing 39 times between 2009 and 2014. This is mainly attributed the various forms of video (TV, VoD, Internet Video, and P2P) that are expected to exceed 91 percent of global consumer traffic by 2014.

The increasing demand for massively distribution and replication of large amounts of resources has led to two main developments: P2P networking and CDNs. Generally, in P2P networking a more scalable distribution is achieved by removing load from origin servers and by leveraging self-organized, adaptive, and fault-tolerant distribution among peers. In many P2P networks, P2P user agents resolve resource names to candidate peers and retrieve fragments (chunks) from selected available peers.

Whereas P2P networking is a specific communication model that is specifically employed by corresponding applications (file sharing, live streaming etc.),

[4] http://www.cisco.com/en/US/netsol/ns827/networking_solutions_sub_solution.html

CDNs are more transparent to users by redirecting requests for web resources to topologically optimal caches – using mechanisms such as DNS-based redirection. In P2P networks, content distribution, local storage of chunks etc. follows a self-organized model (depending on current interest, available resources etc.). In CDNs, content is often distributed administratively onto caches, although it is also possible to respond to dynamic changes of content popularity.

Both approaches represent a move towards a more content-based communication model: URIs and DNS names are interpreted in way that allows accessing cached copies of content (chunks) in the network.

Still, there are number of issues: sub-optimal P2P peer selection that leads to expensive inter-provider traffic – as currently addressed by the IETF ALTO working group; and the inability to effectively leverage in-network storage to reduce overhead for both P2P and CDN scenarios – as currently addressed by the IETF DECADE WG.

In specific network domains, such as 3GPP-defined LTE networks, the currently observed and predicted future increased bandwidth demand has led to architectural changes such as the Selected IP Traffic Offload (SIPTO) concept [7] for reducing load on core network links by replacing inefficient tunneling by more optimal data forwarding. The next obvious step would be to add caching and local CDN caches as well.

Looking at the indisputable 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? Is it possible to develop a general infrastructure that provides in-network caching, distribution of content copies to the right location at the right times?

2.2 Persistent and Unique Naming

When applying the information-centric paradigm today, accessing named content can be challenging, because the Internet, the DNS, and the Web are often used with a locator-based mindset. Most content URIs are actually object locators that, after DNS resolution, exhibit the IP address of a web server that is serving requests by resolving the local part of URIs.

As a result, the name-object binding can easily break, in situations such as:

- object moved within the site;
- object moved to a different site;
- site changed domain;
- site temporarily unreachable (e.g. server overloaded); and
- site permanently unreachable (e.g. company out of business).

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).

In summary, today's Web and CDN architecture treat URIs as object identifiers, but persistent and unique naming is not provided, which leads to the

potential disruption and inefficiencies with respect to object caching and distribution.

2.3 Object Authentication

A partly related problem is the inability to actually verify that some named content is actually what it is supposed to be, which is illustrated by the number of mirror sites for popular software packages that try to convince the user that a specific content is actually authentic, contains no (additional) viruses etc.

Fundamentally the problem is caused by the fact that current authentication technologies such as Transport Layer Security (TLS) and/or Secure Sockets Layer (SSL) provide *connection endpoint authentication* instead of *object authentication*. A retrieved object is considered trustworthy if it can be downloaded over a secured transport connection (with a certificate-backed trust chain).

However, once the object (copy) has left the origin server, its authenticity can normally not be verified, which is a real problem for caching and P2P-based distribution today. Consequently, if in an information-centric approach, it becomes much more important to be able to authenticate objects – and not transport session endpoints.

2.4 Mobility and Multihoming

For managing mobility in today's IP-based networks, there are essentially two approaches for maintaining host reachability: routing and indirection (Mobile IP).

Because of the issues with routing-based mobility (slow convergence, routing state explosion), Mobile IP based indirection is normally considered the practical alternative, although it is generally only deployed in certain networks domains such as mobile communication networks.

Multihoming (attachment to multiple networks at the same time) is considered problematic when IP addresses serve as both host identifiers and host (interface) locators. Host-identity concepts try to de-couple host-identities and network locators, but still require some form of registration/indirection to maintain reachability.

For accessing named content, host identity or even host interface based mobility management appears inadequate: communication endpoint are not interested to establish sessions with specific endpoints – the objective is to obtain a valid object copy from the network, i.e., from a topologically close cache. When nodes lose connectivity, they can re-issue requests at the next attachment point without the need for seamless reachability. Content originator mobility can also be better addressed by object replication to avoid disruptions.

Multihoming, i.e., managing multiple interfaces and communication opportunities, is non-trivial in a locator-based model. Transport layer sessions are normally bound to locators, which makes it difficult to flexibly employ all possible access opportunities simultaneously.[5]

[5]Multi-Path TCP is an attempt to provide some support for such scenarios.

2.5 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 actually not require seamless communication with end-to-end paths [8], and if the primary objective is access to information objects, in-network-caching and/or store-and-forward approaches such as the DTN architecture [9] with its convergence layer concept for hop-by-hop transport could provide better reliability and better performance by leveraging optimized hop-by-hop transport and in-network caching. Corresponding approaches have been described in [10] and [11].

2.6 Leveraging In-Network Caching

Current caching is done either non-transparent (explicit proxy cache configuration on browsers) or transparent (on-path caching based on Deep Packet Inspection (DPI)), but it is not well integrated into the network. Proxy web caches can currently not benefit from P2P caches, local caching on endpoints cannot be leveraged and, recently, there is an increasing trend to disallow caching of objects (Google Map objects, youTube videos) due to DRM concerns.

As mentioned above, the authenticity of cached objects cannot be verified, and the names of cached objects often depend on their locator, which easily leads to duplication.

2.7 Content Distribution Networks

Today, specific content providers use CDNs to accelerate the delivery of content, both static and dynamic. CDNs work by creating an overlay network over the Internet, in order to serve content to clients from the nearest mirror. For live streams, the overlay structure is used to set up a multicast tree to minimize bandwidth consumption and improve performance.

CDN infrastructures today provide sophisticated management tools to configure object placements and path, to manage performance, to assess usage etc. The specific solution is proprietary, i.e., depends on the specific CDN network, which means there is no common approach and of course no interoperability between different systems.

Similarly, it is not possible to share resources between different CDNs nor to inter-connect them, i.e., in-network storage and communication resources are not used efficiently.

3 Generic Building Blocks of an Information-centric Network Architecture

In this section, we will discuss the general ICN concept independent of a specific architectural approach. We will define some common terminology and some building blocks that multiple ICN architectures have in common. Consequently,

not all building blocks described here necessarily have to be part of an ICN architecture.

3.1 Information Objects

First of all, we have to define the meaning of *data* and *information*. In a general definition, Data is the lowest level abstraction whereas information is an abstraction on the next higher level. In order for data to become information, it must be interpreted and must take on a meaning. However, in the following, we will use the terms data and information as well as the term *content* interchangeably unless explicitly stated.

The ICN concept focuses on the information itself and not on the storage location of this information. To highlight this differentiation, we introduce the concept of an *Information Object (IO)*. An IO represents the information itself independent of its storage location and physical representation. An IO can have multiple different *representations*, i.e., unique bit patterns representing the same information, e.g., different encodings. Finally, there can be multiple different *copies* of each representation, e.g., stored on a server, a client node, or in a cache. Consequently, an IO refers to the group of all representations and corresponding copies of the same information. Figure 1 illustrates the relation between an *IO*, its *representations*, and *copies* of the representations.

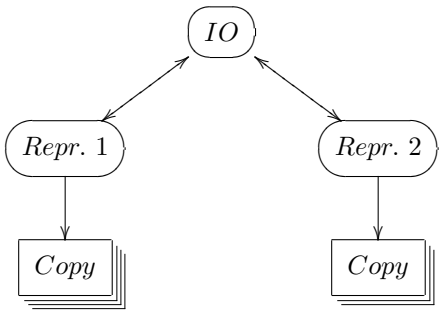


Figure 1: An Information Object (*IO*) with two representations (*reps*), each with multiple *copies*.

There are multiple different ways to implement the IO concept. For example, one ICN incarnation could implement the IO concept as a dedicated data structure containing all *locators* (i.e., addresses) pointing to all respective copies. Such an implementation could additionally contain *metadata*, i.e., data about the represented information, e.g., encoding-related information or security-related information. If IOs are represented by a dedicated data structure, an *information model* is required in addition that defines the syntax and semantics of IOs. In a different ICN incarnation, the IO concept might not be represented by any dedicated data structure at all.

3.2 Naming and Security

Naming information plays an important role in the ICN concept. In today's Internet architecture, we mainly name the storage locations of information, e.g., we use Uniform Resource Locators (URLs) relating to a network node and

file structure to name files, and Internet Protocol (IP) addresses to name the interfaces of the respective storage nodes. In information-centric networks, we name the information itself, i.e., we name Information Objects via *location-independent Object identifier (ID)*.

Naming is closely related to security in several ICN architectures. In today's Internet architecture, security is an add-on to the original architecture that is mainly based on trusting the source of information via authentication and securing the data channel via encryption. In the ICN concept, security cannot be bound to the storage location as the network and/or user should benefit from any available copy. Consequently, new *information-centric security* concepts are required that base security on the information itself. A popular approach followed by several ICN architectures is to integrate security aspects with the naming concept, i.e., the object IDs. We define the following five general technical security goals in the ICN context:

- Confidentiality: Only eligible entities (i.e., users or systems) can read secured information, i.e., IOs and corresponding metadata.
- Data integrity: It is possible to identify accidental or intentional changes to IOs and the corresponding metadata. This is also referred to as *self-certification* when closely integrated with the information itself.
- Accountability: The owner/creator of information can be authenticated and/or identified. We explicitly differentiate between:
  - Owner authentication: Binds the information securely to a virtual entity, represented, e.g., by a pseudonym or a public/private key pair.
  - Owner identification: Binds the information securely to a real-world entity, e.g., a person's unique identity or an institution.
- Availability: The IOs and corresponding metadata published in the network have to be available and accessible for (authorized) entities.
- Controlled access: Access (i.e., read, write, execute) to IOs and/or corresponding metadata can be restricted to authorized entities.

3.3 Application Programming Interface

ICN architectures typically have *information-centric Application Programming Interfaces (APIs)* that are location-independent. The APIs have a *pull-based* interaction pattern, i.e., the user/application requests IOs based on object IDs. Several approaches integrate a *publish/subscribe* component or *event service* component. The resulting major primitives of an information-centric API are of the type *GET(ID)* and *PUT(ID)*. The GET(ID) primitive might involve a direct request of a fully or partly specified object ID, or a subscription for a certain type of information. The PUT(ID) primitive involves some kind of publication and/or registration of new information in the network, depending on the specific architecture.

3.4 Routing and Transport

As the object IDs are location-independent, it is not possible to use common topology-based routing and forwarding algorithms based on these object IDs. There are two general approaches in ICN networks to handle routing, which strongly depend on the properties of the object namespace, mainly if the names are aggregatable or not.

The first approach uses a *Name Resolution Service (NRS)* that stores bindings from object IDs to topology-based locators pointing to corresponding storage locations in the network, i.e., the NRS translates the object IDs into corresponding topology-based addresses. This approach has three conceptual routing phases: (1) routing the request message (i.e., GET(ID)) to the responsible NRS node where the object ID is translated into one or multiple source addresses, (2) routing the request message to the source address(es), and (3) routing the data from the source(s) to the requester. All phases can potentially use different routing algorithms. If the object namespace is not aggregatable, a *name-based routing* method might be used, typically for the first phase. The second and third phase might use, e.g., a common topology-based routing algorithm like IP if the NRS translates the object IDs into topology-based addresses. There are multiple alternatives to loosely or tightly integrate these three phases in an ICN architecture.

The second general approach does not perform any kind of name resolution but directly routes the request message from the requester to one or multiple data sources in the network based on the requested object ID. The routing algorithm used for this approach heavily depends on the properties of the namespace again. After the source has received the request message, the data is routed back to the requester, equaling the third phase in the NRS-based approach. We call this last phase the *data transport* phase. The data transport in ICN networks can rely on one or multiple data sources.

3.5 Caching

To ensure efficient network utilization and improve data availability, several ICN architectures make heavy usage of data *caching*. There are two major caching approaches: *caching at the network edge* and *in-network caching*. Caching at the network edge includes, e.g., user nodes like in P2P networks, and replicated servers. In-network caching describes the caching of data within the transport network, e.g., on the forwarding path in network routers, or in conjunction with the Name Resolution Service.

3.6 Storage and Search

Depending on the actual ICN approach, two additional components that are external to today's Internet architecture might become part of an ICN network architecture. First, persistent data *storage* might be closer integrated with the network architecture, including a closer integration with caching and name resolution. Second, information *search* might be closer integrated than in today's Internet architecture, especially when metadata about IOs is stored in the ICN network.

4 Discussing Information-centric Network Architectures

Based on the identified building blocks in section 3, we will now discuss the instantiation of these blocks for the specific approaches. In subsection 4.1, we first provide an overview of CCN, PSIRP, 4WARD-NetInf, and DONA before we compare with respect to naming/security (subsection 4.2), name resolution and naming (4.3), in-network storage for caching (subsection 4.4), and APIs (subsection 4.5).

4.1 Overview of Information-centric Networking Approaches/Related work

In this subsection we will present the existing approaches to Information-Centric Networks: Content Centric Networking (CCN), Publish-Subscribe Internet Routing Paradigm (PSIRP), Network of Information (NetInf) and Data-Oriented Network Architecture (DONA).

4.1.1 CCN

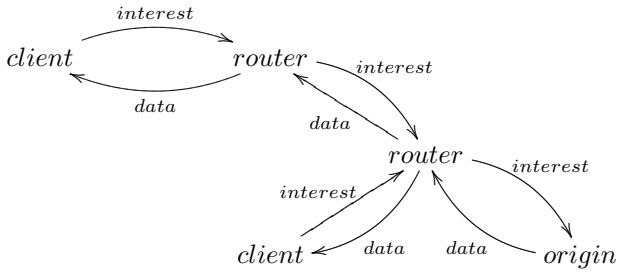


Figure 2: CCN overview

The main idea of CCN is that a request for an information object is routed towards the location in the network where that information object (IO) has been published. At the nodes traversed on the way towards the source the caches of the nodes are checked for copies of the requested IO. As soon as an instance of IO is found (a cached copy or the source IO) it is returned to the requester along the path the request came from. All the nodes along that path caches a copy of the IO in case they get more requests for it.

4.1.2 PSIRP

In PSIRP IOs are published into the network by the sources. Receivers can then subscribe to IOs that have been published. The publications and subscriptions are then matched by a Rendezvous system. The matching procedure results in a rendezvous identifier (RI) that can be seen as an identifier for a communication channel. The RI then, in turn, can be resolved (within a scope) to a forwarding identifier that can be used for routing of data object through the forwarding network.

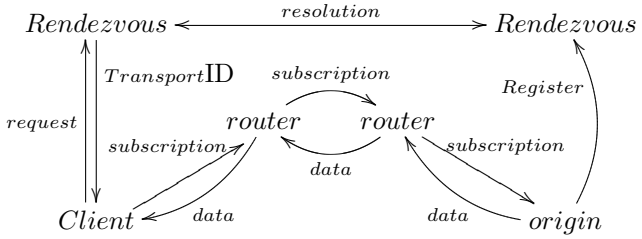


Figure 3: PSIRP overview

4.1.3 4WARD-NetInf

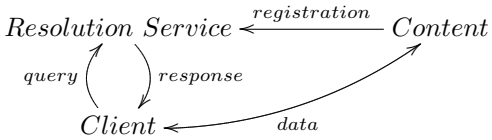
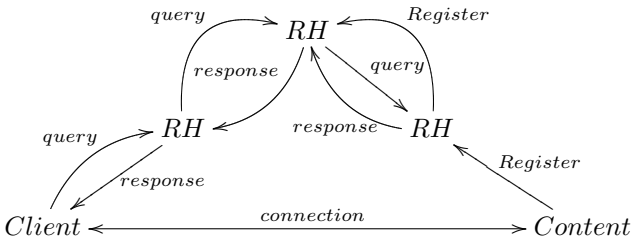


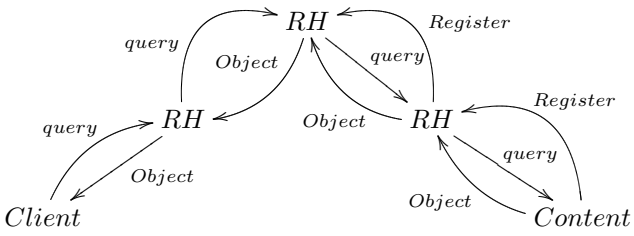
Figure 4: 4WARD-NetInf overview

In NetInf IOs are also published into the network. They are registered with a Name Resolution Service. The NRS also is used to register network locators that can be used to retrieve data objects that represents the published IOs. When a receiver want to retrieve an IO the request for the IO is resolved by the NRS into a set of locators. These locators are then used to retrieve a copy of the data object from the 'best' available source(s).

4.1.4 DONA



(a) IP based connection



(b) Purely data-oriented operation

Figure 5: DONA overview

In DONA IOs are also published into the network by the sources. Nodes that are authorized to serve data register to the resolution infrastructure. Once a given content is registered, requests can be routed to it. Register commands have a TTL, when the registration expires it needs to be renewed. Resolution Handlers have a hierarchical structure. Requests are routed by name in a hierarchical fashion. The resolution infrastructure routes requests by name and tries to find a copy or the content closest to the client. DONA's anycast name resolution process allows clean support for network-imposed middleboxes (e.g. firewall, proxies). In purely data-oriented operation, IOs are routed back along the same path of the request.

Content providers can perform a wildcard registration of their principal in the NRS, so that queries can be directed to them without needing to register specific objects.

4.2 Naming and Security for Information Objects

A naming scheme enabling identification of information objects independent of location is perhaps the most crucial part of an information-centric network design. The design goals for the NetInf naming scheme includes name persistence, self-certification, owner authentication, and owner identification. Name persistence can be maintained even though storage location, the content, the content's owner, or owner's organisational structure change. The separation of self-certification, owner authentication, and owner identification allows the owner to remain anonymous, and allows the publisher to be different from the owner. Both of these properties are important as foundations for trust, privacy, and content management.

With the main design goals being security-related, NetInf names [12] are essentially from a flat namespace. That is, there is not a hierarchical structure that can easily be utilised for, e.g., routing purposes. The type, authenticator, and label fields however provide some structure, and especially the authenticator field, a hash of a public key, could be useful for making name resolution / routing more efficient by aggregation.

The CCN naming scheme is hierarchical in order to achieve better routing scalability through name prefix aggregation. The names are rooted in a prefix, unique for each publisher. The publisher prefix makes it possible for clients to construct valid names for data that does not yet exist, and publishers can respond with dynamically generated data. CCN names are used both for naming information and for transport. The granularity of the names is very fine: single chunks (packets) are named.

PSIRP makes use of two primary namespaces, rendezvous identifiers and forwarding identifiers. Rendezvous identifiers (together with scope identifiers) name rendezvous points which are used to establish contact between publishers and subscribers. There is not a namespace for information objects per se, but the name of the rendezvous point can be used as one. Both CCN and NetInf do it the other way around - information object names can also be used to name rendezvous points (services). In principle, we don't think that there is a fundamental difference. It is more a matter of viewpoint. The second namespace, the forwarding identifiers, is used by the PSIRP forwarding fabric to transport data after contact is established at a rendezvous point. Both namespaces are flat and can be based on the hash of the content or the hash of a publisher public key.

DONA URIs are in the form `P:L`, where `P` is the "principal", which identifies the publisher of the information object, and `L` is the label of the object. Object identifiers are flat and unique in each namespace, and principals are globally unique. Names are inherently secure, as they contain the hash of the public key of the publisher (the "principal" ID).

4.3 Name resolution and routing

Name resolution usually means that a resolution service is queried and one or more locators are returned, which then can be used to retrieve the object. We will in the following call this two-step resolve/retrieve. An alternative is to directly return the object, without first returning locators. We call this one-step, or integrated, resolve/retrieve. Even though the latter often is referred to as name-based routing, name resolution services also often use name-based routing to find the answer to a query. Name-based routing can be described as a way to communicate through the naming layer, rather than through the forwarding layer. We therefore consider name resolution and name-based routing as more or less equivalent functions.

The important differentiator is instead whether the locators are visible to the client or not. The design choice made for the NetInf API was to hide the locators from the client application. This means that both two-step resolve and then retrieve, and integrated resolve/retrieve mechanisms are supported "under the hood".

In NetInf two resolution mechanisms have been developed. One that can do integrated resolve/retrieve called Multi-level Distributed Hash Table [13]. It uses a hierarchy of DHTs for performing the resolution and possibly also the data transfer. There is also a two-step approach Late Locator Construction (LLC) [14] that focuses on handling highly dynamic network topologies, including large moving networks. To make the name resolution robust against large and frequent topology changes locators are constructed at the time of the resolution request.

The advantage of the two-step resolve/retrieve choice is that many different transport mechanisms can be used, including existing ones, facilitating deployment in existing networks. With this model, NetInf has two levels of routing, one for name resolution and one for transport. The integrated resolve/retrieve is on the other hand potentially a simpler and more efficient approach without any dependencies on existing infrastructure.

CCN has an integrated resolve/retrieve mechanism. Clients ask for an information object by sending interest packets, which are routed toward the publisher of the name prefix using longest prefix matching. When a copy of the information object is encountered on the path, a data packet containing the requested object is sent on the reverse path back to the client.

PSIRP has a two-step resolve / retrieve model, where the resolver is called rendezvous point. Forwarding of the data can, as in NetInf, potentially take a different path back to the client than name resolution / rendezvous. The rendezvous point does not have to be on the path to the publisher, nor hold a copy of the data. Data is forwarded using source routing: a Bloom filter describing the route is built by the rendezvous point and used by the requester to reach the destination. The Bloom filter is attached to the packet itself, and it contains all the names of the links that have to be followed.

In DONA, nodes that are authorized to serve data, register to the resolution infrastructure. Only once a given content is registered, requests can be routed to it. A REGISTER command has a TTL: after it expires, it must be renewed. Resolution Handlers (RH) have a hierarchical structure. Requests are routed by name in a hierarchical fashion. Every request that cannot be satisfied is forwarded to the parent RH. The resolution infrastructure routes requests by name and tries to find a copy or the content closest to the client. The resolution infrastructure is meant to work on IP. Once a name is resolved the client connects to the content source. NAT is performed by the caching routers to transparently cache the content.

4.4 In-network storage for caching

In-network storage for caching information objects is a fundamental component of NetInf and other information-centric approaches to networking. In-network storage is one of the distinguishing characteristics compared to overlay and peer-to-peer technology. Without in-network storage, NetInf would not be a network-level technology. The caching is strictly on an opportunistic basis. A cache may delete a cached object at any time. Persistent storage for information objects is something different and is not discussed in this section.

In NetInf there are two ways to make use of a cached copy of an object. Firstly, the copy can be found directly by querying the name resolution system, provided that the copy is explicitly registered there, or provided that the copy at query time is found by a name resolution system based on local search (e.g., broadcast). Secondly, the copy can be found by a cache-aware NetInf transport protocol on the path to a location known to hold a copy, for example, a location retrieved from the name resolution system.

CCN routes a request for data towards the publisher, and makes use of any cached copies along that path. Copies can also be found by local search. These cases are functionally fairly equivalent with NetInf. Since CCN transport and name-based routing are integrated, there is no separate case for CCN transport. It is to be noted that in CCN atomic objects are single packets, so 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.

In DONA, caching is inherent in the architecture. Any cache can respond to a FIND request and serve the relative IO.

4.5 Application Programming Interface

The APIs used in exising approaches have usually just a few functions, mostly related to name resolution/registration and object request forwarding. Tables 1, 2, 3 and 4 summarize the key functions for each approach.

common API		
`ccn_connect`		connect to the local CCN service
`ccn_disconnect`		disconnect from the local CCN service
sender API		
`ccn_put`	(PUT)	send a data obejct
`ccn_set_interest_filter`		create a filter for receiving specific interests
receiver API		
`ccn_express_interest`	(GET)	express an interest
`ccn_get`	(GET)	receives a data object
`ccn_verify_content`		verifies an object

Table 1: CCN API

sender API		
`publish`	(PUT)	publish an object
`revoke`		revoke a previous publication
`send`		stream data to a channel-type connection
receiver API		
`resolve`	(GET)	resolve a name into a list of metadata descriptors (incl. locators)
`join`		connect to a service
`receive`	(GET)	receive an object or a stream from a connection

Table 2: NetInf API

sender API		
`REGISTER`	(PUT)	register an object to the name resolution system
`UNREGISTER`		unregister an object from the resolution system
receiver API		
`FIND`	(GET)	obtain the object corresponding to the given name

Table 3: DONA API

common API		
`attachPolicy`		attach a policy to the given scope or rendezvous ID, without needing to resubscription
`associate`		associate a given rendezvous ID to a given scope
sender API		
`forward`		send data to a receiver
`register_provision`		used when a publisher is ready to commence transmission to subscribers and requires forwarding information to do so
`withdraw_provision`		withdraw a standing provision
`publish`	(PUT)	publish an object
`advertise`	(PUT)	advertise a given object
receiver API		
`listen`		receive data from a stream
`mute`		stop receiving data from the stream
`register_interest`	(GET)	used by a subscriber when it is ready to start receiving data.
`withdraw_interest`		withdraw a standing interest
`subscribe`	(GET)	subscribe to a given rendezvous identifier
`unsubscribe`		unsubscribe from an active subscription

Table 4: PSIRP API

For each approach, functions are separated into *sender*, *receiver* and *common*, according to the use they are intended.

In each table there is a reference to the relevant abstract ICN APIs, as indicated in section 3.3.

5 Properties and Challenges

In this section, we discuss a few properties and challenges for ICN as identified by the 4WARD project. We discuss scalability issues (subsection 5.1), security (subsection 5.2), privacy (subsection 5.3) and mobility (subsection 5.4).

5.1 Scalability

Technology for world-wide communication networks have to be designed and evaluated with respect to scalability that meets current and future foreseen requirements for network size and performance. Scalability and performance often goes hand-in-hand – performance usually decreases with larger network size.

One of the main scalability issues in current IP networks is the number of IP address prefixes in the backbone routing tables. For information-centric networks, the main scalability issue is instead the number of information objects, or more precisely, the amount of bookkeeping needed to keep track of all information objects.

So how many objects can we expect? In summer of 2008, Google reported that they hit the milestone of one trillion (1,000,000,000,000) unique URLs on

the web[6]. Others estimate the number of web pages to at least 18 billion[7]. Another measure of interest to our discussion is the number of second-level domains registered in the Domain Name System, which currently (November 2010) is about 118 million[8].

Empirical evidence shows that it is possible to operate a DHT 24x7 that at any given time consists of more than 2 million nodes [15]. Assuming two million nodes for the resolution infrastructure is therefore not unreasonable. We also set the target number of objects that needs to be supported to 10^{15}, three orders of magnitude larger than the number of unique URLs mentioned above. With no replication and 100 bytes per resolution record, each node needs 50 GB of memory. With 10 times replication and 1 KB per record, we instead need 5 TB of memory for each name resolution node. 50 GB of memory can be implemented with DRAM technology, but also easily supported with SSD technology. 5 TB is feasible with state-of-the art SSD storage technology, but admittedly currently somewhat expensive. But as SSD technology is developing rapidly, we are confident that this will not be a big issue in the near future. We estimate that the lookup times within one node with SSD storage will be less than 1 ms.

NetInf's flat information object namespace means that the NetInf name resolution system needs to support one entry per information object, clearly a huge number as discussed above. If the routing can be aggregated on the authenticator label in the NetInf names, the bookkeeping is reduced, perhaps to the level of the number of second-level domain names. Investigation on the performance of KAD-based resolution [16, Sec 3.3] indicates that total lookup latencies in the order of 100ms can be achieved at the global level.

Object names in CCN are by design hierarchical. The CCN global routing system needs at least to be able to handle name prefixes at the level of publishers. The current number of second-level domain names mentioned above is therefore a reasonable estimate for the number of such prefixes. Another scaling issue in CCN is the per-packet forwarding state needed along the end-to-end path to guide data packets back to clients. On the plus side, per interest packet allows for multicast without additional infrastructure.

In PSIRP, the use of both rendezvous and scope identifiers makes a two layer structure possible such that a flat rendezvous name space can be handled in separate scopes; the scopes have no fixed relationship or structure. The registration of rendezvous and scope identifiers is therefore scaling based on local rendezvous locator knowledge within one scope. The location of scopes can be compared to DNS entries, such that the number of scopes can at least be as many as what DNS serves at the moment.

PSIRP has a two level name hierarchy with the scope and rendezvous identifiers. Rendezvous identifiers are registered within a certain scope. The scopes have no fixed relationship or structure, which means that you first need to find the correct scope in order to find a certain rendezvous point. The scopes can be compared with the publisher prefixes of CCN and the authenticator label of NetInf. Above we estimated them by comparing with the number of second-level domains in DNS. The scopes can also be compared with the Autonomous Domain (AS) numbers of the Border Gateway Protocol (BGP) which are used

[6] `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`
[7] `http://www.worldwidewebsize.com/`, Dec 3, 2010
[8] `http://www.webhosting.info/domains/global_stats/total_domains/`

to implement global routing in the Internet. Currently there are a little more than 60.000 AS numbers allocated of which about 34.000 are announced in BGP.

Stateless Bloom filters are used for multicast packet forwarding in PSIRP. It works well for a small group of recipients, but does not scale to groups with a large number of recipients. The step between the rendezvous point and the Bloom filter packet forwarding is done with a "topology formation" (TF) function. As the TF has not been provided it is challenging to understand system scalability issues.

In DONA, the resolution layer is based on a hierarchical NRS. Global scalability is achieved in the same way as DNS.

5.2 Security

NetInf provides mechanisms for author authentication and origin verification, as well as mechanisms for checking content integrity; the mechanisms are integrated into the network service. These functions are non-existent in today's network where security is mostly based on trusting the server delivering the information.

The NetInf naming scheme provides self-certification of the integrity of the information object data. Self-certification means that no third party or PKI system needs to be consulted, and thus that the integrity of the data can be verified off-line. This property is achieved by providing a signature of the hash of the content as metadata to the object, and including the cryptographic hash of the public key corresponding to the signing key in the authenticator field of the identifier itself. In this way, the object identifier is securely bound to the object data. Owner authentication and owner identification is achieved also with signatures, but for the latter trust in the signing key needs to be established using additional means, for example, a certificate-based PKI.

Due to the security design goals of the NetInf naming scheme, more precisely the self-certification, the object identifiers are not very human friendly, as they consist of more or less random bit-strings. This deliberate trade-off means that additional means are required to securely bind more human-friendly application-level names to the NetInf identifiers. An approach could be to supply signed human-friendly names as metadata to objects. That would at least provide a binding from the object to a human-friendly name.

CCN names are hierarchical starting with a publisher prefix followed by a pathname in a similar fashion as current URLs. While CCN names can be human-friendly, trust in the signing key always needs to be established using external means, including for checking data integrity, since there is no direct binding between the content and the name.

PSIRP has direct security properties with rendezvous and scope identifiers built from hash of content, and indirect security properties through the hash of publisher key plus a label. PSIRP thus has the same issues with human-readable names as NetInf does. To combat DoS attacks, a packet-level authenticator is added to each packet.

In DONA, names include the hash of the public key of the publisher (the "principal"). This means that if the same content is to be published by different publishers, it will have multiple names. This is not really a big problem since it is possible to perform wildcard queries, ignoring the principal, and it's possible for publishers to delegate to other keys the authorization to publish with the

same principal. DONA has the same issues with human-readable names as NetInf and PSIRP.

5.3 Privacy

By making information visible and identifiable at the network level, it is easy to claim that information-centric networking designs could threaten freedom of speech, or at least provide a more effective tool for operators to filter, from their perspective, undesired content or services. "Undesired" could mean illegal due to copyright issues, that the operator wants to differentiate its service (c.f., the network neutrality debate), or that a government wants to suppress freedom of speech.

Since NetInf naming supports unique signing keys per object, resulting in unique authenticator identifier fields, the publisher can prevent easy filtering of all objects from itself. The question is if operators will allow unique keys. Perhaps operators will charge for each key used to publish content, especially if the routing system will need to be able to aggregate based on the authenticator field.

But, in the end, is there a principal difference compared to today? Most user generated material is published on a site that is controlled by a big actor (e.g., YouTube), a situation which is likely to remain. Information object identification will here not add much of filtering capabilities, since the object identifier most likely will be created using a key of the big actor, and not of the owner. The conclusion is that filtering will be about equally easy or hard as filtering on IP addresses and using packet inspection.

The only way to control who is downloading any specific content in CCN is to control all the last-hop routers, which can mean every single router in the network.

5.4 Mobility

In NetInf, mobility is achieved in a way similar to mobile-IP: there is a central rendezvous point for every end-user equipment. Seamless handover of client is possible. Content-provider mobility is also possible, but not seamless.

In PSIRP, client can just unsubscribe, switch networks and resubscribe again. A new path/subtree will be computed by the routing layer. Bufferuing and sequence numbering allow for seamless handovers. Content provider mobility is more complex, and involves updating the routing state in the rendezvous nodes. We don't know exactly how that is supposed to be done, but that process probably has a slow convergence.

Client mobility in CCN is inherent. A client can just 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 less easy: a content-provider would have to update the routing tables of all the relevant the neighbouring nodes, which also is slow. Furthermore, many moving content-providers would polute the routing tables with specific prefixes, countering the advantages of prefix aggregation.

Client mobility in DONA is achieved in a way similar to PSIRP: clients can de-register from their previous location and re-register at the new location. De-

registration is not mandatory, as Resolution Handlers can expunge the entries regarding content they could not find.

6 Deployment Issues

The different information centric approaches presented in this paper are more or less taking a clean-slate approach to designing the network of the future. However for this research to make it over the divide between interesting academic research and into real world deployment we need to consider a migration path from today's networking to the new ICN technology. In this section, we therefore discuss incentives for different actors on the networking arena to deploy ICN technology.

We consider five types of actors: (1) end users, which can be a private person, or an organisation using a network service; (2) access network operators that provide network access service to end users; (3) connectivity network operators that provide connectivity for other operators; (4) content or service providers, that have content that they make available, possibly against payment, to users; and (5) application developers who should benefit from direct access to information objects without having to be concerned with networking details.

Admittedly, these definitions are deliberately stereotyped and merely serve to make the following discussion a bit clearer. We have no intention to succinctly define the edge cases, for instance, end users providing network access to others, or operators that are both access providers and provide connectivity for other operators.

The main issue impeding deployment is that one or more of these actors need to be given incentives to start deployment of the core ICN functionality. This functionality boils down to two main network components: in-network storage for caching of information objects, and the name resolution/routing service.

6.1 Incentives for end users

Current peer-to-peer overlay technology implement a similar service as what ICN technology can do. There are peer-to-peer based content distribution networks where the users provide storage and upload capacity for the benefit of other users. We therefore conclude that it seems to be very likely that end users also have incentives to deploy ICN technology. End users with personal computers commonly already have the storage and processing resources needed to implement the functionality needed for ICN, so there is no costly investment needed.

We however argue that end-user deployment is not enough. If only end users deploy ICN, there is not any gain beyond what peer-to-peer and overlay technologies can provide. To get the full potential, network providers have to be involved. An exception is the local collaboration use-case, where you can get the major benefit of ICN with only end-user deployment.

6.2 Incentives for access network operators

Access network operators get their revenue from the end users they serve. They have to pay connectivity network operators for the traffic their users generate to

and from the rest of the world. The amount depends on the capacity of the link and/or the volume of traffic in both directions. With these prerequisites, the access network operator has incentives to reduce the volume of traffic per end user to/from other operators, since that will both reduce the investment needed per user and reduce the traffic cost per user. We therefore conclude that access network operators seem to have incentives to deploy ICN technology because its caching will reduce the volume of traffic to/from other operators. There is a clear cost incurred by this deployment, since storage available for ICN caching is not part of the network equipment that access operators currently have. The question is whether this investment cost is low enough to motivate deployment. We believe that access operators are the critical actors. If they do not deploy ICN technology, we will not do better than current peer-to-peer technology.

We can also compare with the Akamai content distribution network. As we understand, they are making agreements with access network operators to deploy their CDN storage within the access operator's network. Neither Akamai nor the network operator pays to the other, so there must be benefits for both by this arrangement. The difference compared to the ICN case is that the network operator does not need to take the investment cost, but instead Akamai does. As Akamai charges the content providers for their caching service, the interesting question for ICN is whether the access operator also needs to charge in some way for their ICN caches in order to motivate the cost of deployment. See [17] for a more comprehensive survey of CDNs.

Another comparison is with the currently ongoing discussion in the IETF ALTO working group. The objective there is to reduce inter-operator traffic induced by peer-to-peer applications by providing a peer selection mechanism that favours peers located within the same operator network (Autonomous System - AS). Since ICN caching also has the ability to localise traffic within the operator's network, provided that there is incentive for deploying ALTO, by conjecture there is incentive to deploy ICN. A similar argument can be made for the storage needed by the functions that the IETF DECADE working group will develop. But this argument is stronger, since DECADE needs storage which has higher investment costs than the ALTO mechanism.

6.3 Connectivity network operators

Connectivity network operators get their revenue from delivering others' traffic. They normally charge based on the amount of traffic received from other connectivity network operators, or the total traffic exchanged with access network operators. In both cases, connectivity network operators have incentives to increase the amount of traffic. If such an operator deploys a (ICN) cache, their revenues will likely decrease. With this reasoning, incentives work against ICN deployment.

This one example that there might be a need to introduce new business models in conjunction with introducing ICN. With ICN, traffic is not pushed through the network but generated in response to requests. It thus seems to make sense that providers able to respond to these requests should be rewarded, not penalized.

6.4 Content / service providers

Content and service providers want their content and services to be made available for their users. They sometimes charge the users for this, sometimes they get their revenue from advertisement. Big content providers today have to pay Akamai or similar content distribution networks for making their content available in large scale. In-house equipment simply does not suffice for popular sites and content. The content providers are thus indirectly paying for the investment in CDN caches within the access operator's networks. Content and service providers would certainly not mind if CDN functionality would be built into the network, as would be the case with ICN. They will not need to use a CDN network, and will thus save that cost. It is however clearly infeasible for a content provider to have agreements with all access operators for using their caches to distribute the content. The situation with ICN therefore potentially increases the current tussle between network operators and content/service providers where the former group wants a share of the latter's revenue in order to finance the investment in network equipment. A different business model compared with today might be needed to resolve this issue. ICN might be an opportunity to introduce new business models that resolve the network operator / service provider tussle. Additionally, for content/service providers to be happy with a ICN caching solution it is important that ICN can provide appropriate feedback on how frequently and by whom content is accessed in the caches.

6.5 Application developers

For application developers an ICN API which provides direct access to the information objects needed for the application should be of great interest. Especially if it provides them with the same or better efficiency than today's CDN and p2p solutions do.

In addition an application independent ICN API should make it easier to reuse information elements created by other applications in new applications. This should cater for a more vital application ecosystem.

7 Remaining challenges

The analysis of current ICN research so far leads to a set of remaining challenges and research topics that we summarize in this section.

7.1 Naming

The specific approaches taken for naming have implications, for instance while NetInf provides a secure, flexible and extensible naming system, the names are not human readable, which of course goes for many URLs today as well. There are a lot of other alternatives for requesting objects in a human-friendly way such as selecting alternatives from menus, icons, results from search services, etc. But for communicating references to objects directly between humans (e.g. when talking to each other) human readable/speakable names are still useful. For the user that means that a directory/search service or some type of application support is needed to map human understandable concepts to NetInf IDs. A

critical issue is how this mapping can be made secure so that attacks are not possible.

In general, a clear understanding of the purpose of a naming system is important. If the primary purpose is unique and persistent object identification, it may not be required (nor desirable) to use the same names for constructing hypertext webs. On the other hand, adding another layer of indirection can create lookup inefficiencies and also ambiguities regarding using these layers for linking between objects. In the Web today, it is possible to link to an object using both DNS names or IP addresses in the URI – if the DNS name revolves to the same IP address, both URIs could link to the same object. Similar properties would be desirable for a NetInf-based web as well.

Routing in CCN is performed on names. CCN names are hierarchical, and each component is an arbitrary sequence of bytes, so CCN names can potentially (but not necessarily) be human readable. The biggest downside of routing by name is fragmentation of routes. If every name can be independently routable, routing table in routers may become quite large. A possible solution for that problem is to use the CCN names for routing, adding a human-readable naming layer on top. This is the approach proposed for Named Data Networking (NDN), but it required similar considerations as described above for flat names: namely securing the mapping between human-friendly names and CCN names.

NetInf and CCN naming includes a versioning concept. The best ways of using this versioning concept needs further study. To study alternative ways of dealing with versioning can of course also be of interest.

PSIRP naming is organized on multiple layers, the topmost of which is the "Application identifiers" layer, where applications can decide their own names. This system suffers from requires the same security concerns as the NetInf solution.

DONA names are flat, so they too have the same problems as NetInf names. Routing in DONA can be completely name based.

7.2 Name resolution

The NetInf architecture is open such as multiple NRSs can be used. By making the IOs self-certifying we have made it possible to use NetInf without having to have a trust relationship established with the NRS.

By the fact that we have multiple copies of IOs and that there can be different versions of the IOs, there can exist a certain uncertainty of the status of an IO, especially considering the use of multiple NRS and the possibility of a partitioned network. We have therefore discussed the possibility of introducing an authoritative home NRS for each IO. If it is crucial to get the latest version of an object, not only the 'best' available copy the authoritative NRS should be used to resolve the NetInf ID. The home NRS could be indicated in the metadata, or it can be determined by e.g. looking up a part of the ID, depending on the chosen name structure. Another aspect that might require a trusted NRS/trusted server is key revocation, depending on how you handle key revocation.

Names in PSIRP are registered at the home rendezvous point. In the case of DONA, Names are registered in the DONA NRS; Resolution Handlers know the location of the IOs and can either return a locator (not visible to the application) or the object itself.

Other aspects that apply to NetInf, DONA and PSIRP include the security and robustness of NRS. How susceptible they are to Denial of Service (DoS) attacks etc. What management of NRSs is needed, e.g. maintenance of bindings, refresh mechanisms, garbage collection, etc.

In CCN, names are not resolved (by a network resolution service), so the concerns mentioned above do not apply to CCN.

7.3 Routing, Topology Changes and Mobility

Although the general idea of ICN is to go beyond locator-based addressing, there has to be some relation or mapping from the name-based topology to the organizational/physical network topology. In the end, content needs to be located on some physical origin server or cache.

In a network with a static topology, approaches such as CCN's hierarchical naming scheme seem attractive. Names can be structured according to the organizational topology such as `net/isp1/com/example/media/video1.mp4`. Sources can register such names with the network, and the network can aggregate such names to enable efficient routing. For mobile clients, there is also no problem – they can attach to different access networks and re-send interest packets as required.

The bigger problem is source mobility: if the responsible node for `media/video1.mp4` moves, it can re-register using the fully qualified resource name – which would however jeopardize aggregatibility and scalability of the routing system. The other alternative would be re-naming – which would give up name persistence.

Adding another layer of indirection, i.e., a naming layer with persistent object names that are mapped to more topologically relevant names, can be a solution – in fact this is one of the proposed directions for the NDN project.

In a name-resolution-based approach, agility with respect to mobility and topology changes depends on design of the name-resolution system. It seems challenging though to solve "seamless reachability" of fast moving origin servers or caches by name resolution.

7.4 Disruption Tolerance

Network disruption is a particular form of topology change that incurs additional challenges. In particular, network disruptions can make name resolution either fail (e.g., when elements of a distributed resolution services are affected by network partitioning) or become unreliable (e.g., when names are resolved to locators that are not usable anymore – i.e., the resolution system has stale information for disconnected areas).

Research on DTN [9] has addressed similar problems. A generally useful approach seems to be some form *late binding*, where names are resolved at the latest opportunity – i.e., name-based routing, perhaps leveraging a hierarchical topology-mapped naming scheme, can be helpful in such scenarios.

7.5 Network Heterogeneity

Related to disruption and delays, it seems generally challenging to design a general ICN system that works well and performs well across different types

of networks, e.g., well-connected infrastructure and DTN type networks. Some issues are similar to today's Internet – not every transport protocol works well end-to-end in the presence of challenged network links. Other issues such as the above-mentioned problem of name resolution are rather specific to ICN.

One possible approach would be to design some support for heterogeneity into a general ICN architecture, i.e., support for domain-specific name-resolution, forwarding strategies, routing, and transport protocols. To this regard, some elements from the DTN architecture in addition to late binding concepts could be useful such as the concept of convergence layers that enable employing specific lower layer hop-by-hop transport as required.

7.6 Scoping of IOs

It is clear that not all objects should be available to everyone. By encryption and access control mechanisms it is possible to restrict access to the content of objects to those that should rightfully access objects. But sometimes the mere existence of an object might reveal more information than the owner of the object wants to reveal to the surrounding world.

To avoid this, a NRS for a NetInf-like approach should have a way to decide how to respond to a resolution request for a certain NetInf ID. One way to achieve this is to introduce a way for the publisher of an IO to add a scope for the registration of the IO into the NRS. The scope would then restrict which NetInf users/applications/objects that will get a response to their resolution request for a certain object, others will simply get the response that the object does not exist.

Another reason why scoping of NetInf IDs might be a good idea is the issue of scalability of the NetInf NRS. By scoping NetInf IDs it would be possible to restrict to what extent a certain NetInf ID will propagate throughout the NRS.

Scoping in DONA and PSIRP is built-in, so no additional work is needed.

8 Conclusions

In this paper we have presented an overview of the research field Information-Centric Networking. The overview discusses the problem statement ICN is addressing, including the issues of scalable and cost-efficient content distribution, the need for persistent and unique naming of information objects and how objects can be authenticated without the need to trust the infrastructure that delivers them.

Taking the issues in the problem statement as a starting point we then outline a set of generic information-centric building blocks that can be used to design an architecture that fulfils the set of requirements that can be derived from the presented issues. The most fundamental building block is of course the information object itself. A secure naming system for the IOs is critical as well as an application friendly publish/subscribe API for accessing them. Storage and caching of IOs in an ICN network are also key components as well as the routing and transport protocols needed.

Four ICN approaches, NetInf, CCN, PSIRP and DONA are briefly described and analyzed with respect to how they implement the generic building blocks that the paper has introduced. The general challenges of scalability, security,

privacy and mobility that all ICN networks face are discussed. As solving all the technical challenges is not enough to ensure the success of a new technology on the market a set of deployment issues are discussed. Which are the incentives to deploy ICN for the different players on the market, end-users, network operators, content/service providers and application developers?

Finally a set of remaining challenges and outstanding research challenges are outlined.

9 Acknowledgments

The authors would like to thank the partners from the 4WARD and SAIL projects for their contributions in many discussions on ICN.

References

[1] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: Line Speed Publish/Subscribe Inter-networking," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. New York, NY, USA: ACM, 2009, pp. 195–206. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1592568.1592592

[2] B. Ahlgren, M. D'Ambrosio, C. Dannewitz, A. Eriksson, J. Golić, B. Grönvall, D. Horne, A. Lindgren, O. Mämmelä, M. Marchisio, J. Mäkelä, S. Nechifor, B. Ohlman, K. Pentikousis, S. Randriamasy, T. Rautio, E. Renault, P. Seittenranta, O. Strandberg, B. Tarnauca, V. Vercellone, and D. Zeghlache, "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/.

[3] B. Ahlgren, M. D'Ambrosio, C. Dannewitz, M. Marchisio, I. Marsh, B. Ohlman, K. Pentikousis, R. Rembarz, O. Strandberg, and V. Vercellone, "Design considerations for a network of information," in *Proceedings of ReArch'08: Re-Architecting the Internet*, Madrid, Spain, Dec. 9, 2008.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/1658939.1658941

[5] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proceedings of SIGCOMM'07*, Kyoto, Japan, Aug. 27-31, 2007.

[6] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing," RFC 4984 (Informational), Internet Engineering Task Force, Sep. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4984.txt

[7] 3GPP, "Local IP Access & Selected IP Traffic Offload (LIPA-SIPTO)," 3rd Generation Partnership Project (3GPP), TR 23.829, Sep. 2010. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/23829.htm

[8] J. Ott and D. Kutscher, "Why Seamless? Towards Exploiting WLAN-based Intermittent Connectivity on the Road," in *Proceedings of the TERENA Networking Conference, TNC 2004, Rhodes*, June 2004.

[9] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," Proceedings of ACM SIGCOMM 2003, Computer Communications Review, Vol 33, No 4, pp. 27–36, August 2003.

[10] J. Greifenberg and D. Kutscher, "Efficient publish/subscribe-based multicast for opportunistic networking with self-organized resource utilization," in *The First IEEE International Workshop on Opportunistic Networking*, 2008.

[11] G. Sollazzo, M. Musolesi, and C. Mascolo, "Taco-dtn: A time-aware content-based dissemination system for delay tolerant networks," in *MobiOpp 2007 Workshop*, 2007.

[12] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, "Secure naming for a network of information," in *Proc. 13th IEEE Global Internet Symposium 2010*, San Diego, USA, March 2010.

[13] M. D'Ambrosio, P. Fasano, M. Marchisio, V. Vercellone, and M. Ullio, "Providing data dissemination services in the future Internet," in *World Telecommunications Congress (WTC'08)*, New Orleans, LA, USA, Dec. 1-2, 2008, at IEEE Globecom 2008.

[14] A. Eriksson and B. Ohlman, "Scalable object-to-object communication over a dynamic global network," in *Proceedings of Future Network and MobileSummit 2010*, June 2010.

[15] M. Steiner, T. En-Najjary, and E. W. Biersack, "A global view of kad," in *ACM SIGCOMM Internet Measurement Conference (IMC 2007)*, San Diego, CA, USA, Oct. 24-26, 2007.

[16] B. Ahlgren, M. D'Ambrosio, C. Dannewitz, A. Eriksson, J. Golić, B. Grönvall, D. Horne, A. Lindgren, O. Mämmelä, M. Marchisio, J. Mäkelä, S. Nechifor, B. Ohlman, S. Randriamasy, T. Rautio, E. Renault, P. Seittenranta, O. Strandberg, B. Tarnauca, V. Vercellone, and D. Zeghlache, "Netinf evaluation," 4WARD EU FP7 Project, Deliverable D-6.3, Jun. 2010, fP7-ICT-2007-1-216041-4WARD / D-6.3, http://www.4ward-project.eu/.

[17] A.-M. K. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks."