From ID/locator split to ICN

Börje Ohlman
Ericsson Research

***Abstract*—In this paper we give some background to how the long-lasting discussion on separation of identifiers and locators in networking has influenced the Information-centric Networking (ICN) research. We start with a historical retrospect of the id/locator discussion. We then address the problems that have been fueling the discussion followed by a chronological walk-through of a number of id/locator split ideas up to some current ICN approaches. Finally we discuss a set of remaining research challenges that need to be addressed before ICN could become a way to resolve the id/locator issue.**

I. Introduction

This paper is intended to give some background to how the long-lasting discussion on separation of identifiers and locators in networking has influenced the Information-centric Networking (ICN) research[1]. We also provide some introduction to ICN in Section II. For a comprehensive overview of ICN and its approaches please refer to [1].

There are some recurring issues with the original design of the Internet architecture that never seem to be put to rest. After the too short address field in IPv4 many people view the decision to not make an explicit separation between network attachment points and node identities (this was part of one of the early proposals) as the major shortcoming of today's Internet architecture. Over the years there have been many proposals how true id/locator separation can be achieved in the Internet. Motivations for introducing id/locator separation have varied over the years and include mobility, QoS, security, naming/addressing and multipoint connectivity.

In 1978 an id/locator split was was proposed to resolve was the issue of separating the naming of hosts from topological network attachment points [2]. This is something that IP fails to do by using IP addresses both for routing to network locations and to identify interfaces of hosts.

Another early id/locator split proposal, in 1991, was made to be able to use of multiple addresses for the same name/content to be able to do policy routing [3], i.e. to be able to choose between alternative paths with different characteristics and or cost.

In 2000 some first ideas to name information objects persistently with names was proposed [4]. The authors also introduced the idea of name-based routing. The next commonly addressed topics were mobility and multi-point connectivity. As security became a more urgent matter the need for stable endpoints increased.

The introduction of the web and peer-to-peer application has made it clear that what most people are interested in when they are using the Internet is not to route to network locations, nor to connect to specific hosts, but to get hold of specific Information Objects (IO)[2] . Which host that delivers the IO from which network location is not of interest to most users, nor are they aware of it. This is how CDN and peer-to-peer applications like BitTorrent works.

This combined with the ideas of using flat labels and potentially also being able to route on them we believe are cornerstones that have lead to today's ICN concepts.

To provide an application independent network service that integrates the success of CDN and peer-to-peer into the network seems like a compelling idea. However, besides the deployment issues there are still major research challenges that remain to be resolved. One of them is naming and routing. It very much relates to the id/locator issue that we are discussing in this paper. A key routing issue is to be able to scale to 10^{15} objects, which is the order of magnitude we think ICN will need.

In addition we note, that for ICN the basic question for routing, *How do I get A from location X?* changes to *Where can I find a copy of A?* This means that the focus for naming and routing is shifting from establishing an end-to-end connection to a specific host to making anycast requests for a named object without really knowing from where it will be delivered.

II. ICN Introduction

Traditional host-centric networking (HCN) is focusing on connecting nodes in the network, primarily via point-to-point connections using e.g. TCP. The most common use of the Internet today is to retrieve content. For this we use the web, CDN and peer-to-peer (P2P) applications. These functionalities are today provided as application-specific overlays on top of the global IP network.

Information-centric Networking (ICN) is focusing on the information objects themselves. The ICN architecture integrates CDN and P2P functionality into a new application-independent network service. Key ICN principles include, naming of content by globally unique identifiers, extensive use of in-network caching, object security and being request driven, including request aggregation.

In HCN the sender sends data to a receiver over a point-to-point connection that goes end-to-end. In ICN the publisher publishes Named Data Objects (NDO) under globally unique names. The NDOs are then requested by those that want to receive them. The NDO requests are handled in the network

[1]We are well aware that our coverage of ideas and approaches will be far from complete, but we hope our selection represents the discussion over the year, other authors would most likely made different selections.

[2]We will in use three terms with roughly the same meaning: Information Objects (IO), Named Data Object (NDO) and Named Object (NO). IO is used when we are talking of naming an object with primarily a semantic meaning. NDO is used in ICN specific contexts. NO is the term we propse to use when formally discussing id/locator separation.

in a hop-by-hop fashion. At each hop when an ICN router receives a request for an NDO it first checks if it is available in the local cache; if found, it returns the content to the requester. If the NDO is not found in the cache the router will check if it already has forwarded a request for that object. If so, it notes that when the object is received it should be returned also to that requestor. For a new request, the router decides on which interface(s) to forward the request.

In ICN it is possible to retrieve NDOs from any cache in the network. This requires a new security model. Today we trust a host and set up a secure tunnel to that host. With ICN we will get objects from untrusted sources through untrusted connections. This requires a strong secure binding between the content and the name of an object. This can be provided by using the hash of the content as the name of the object or by use of publisher signatures.

III. The original idea

Now let us go back to the basics of identifier/locator separation. In 1978 in IE Note #19, "Inter-Network Naming, Addressing, and Routing"[3], John F. Shoch, proposed the following general definitions: a *name* identifies what you want, an *address* identifies where it is and a *route* identifies a way to get there.

In his 1982 paper "On the Naming and Binding of Network Destinations"[5] J. Saltzer (Also published as Internet RFC1498 in 1993) gives the following reflection on this:

> There will be no need to tamper with these definitions, but it will be seen that they will leave a lot of room for interpretation. Shoch's suggestion implies that there are three abstract concepts that together provide an intellectual cover for discussion. In this paper, we propose that a more mechanical view may lead to an easier-to-think-with set of concepts. This more mechanical view starts by listing the kinds of things one finds in a communication network.

He continues:

> In a data communication network, when thinking about how to describe the destination of a packet, there are several types of things for which there are more than one instance, so one attaches names to them to distinguish one instance from another. Of these several types, four turn up quite often:
>
> 1) Service and Users. These are the functions that one uses, and the clients that use them. Examples of services are one that tells the time of day, one that performs accounting, or one that forwards packets. An example of a client is a particular desktop computer.
> 2) Nodes. These are computers that can run services or user programs. Some nodes are clients of the network, while others help implement the network by running forwarding services. (We will not need to distinguish between these two kinds of nodes.)
> 3) Network attachment points. These are the ports of a network, the places where a node is attached. In many discussions about data communication networks, the term "address" is an identifier of a network attachment point.
> 4) Paths. These run between network attachment points, traversing forwarding nodes and communication links.

Then in 1991 Paul F. Tsuchiya in "Efficient and Robust Policy Routing Using Multiple Hierarchical Addresses"[3] comments on Shoch's definitions:

> Briefly stated, the problem with this "where" (addressing) notion is that 1) we tend to think of things as being in only one place at a time, implying that we need only one address at a time, and 2) since we relatively rarely move our computers or telephones, we tend to think of addresses as rather static.

He points out that e.g. Ethernet addresses are completely flat, from a routing perspective, and do not contain any routing information. At the other end of the spectrum there are source routes. If a source route is provided in the address field of a packet there is no need for any additional routing information such as e.g. routing tables. He concludes: "a hierarchical address is nothing more than a special type of source route." He then goes on to say:

> ...I think the practical considerations of routing indeed have to do with what is done in the packet header and what is done in the routing tables. Therefore, I choose to further classify along those lines, and partition the routing function into header routing and table routing.

Paul F. Tsuchiya also provides a classification of header routing. He divides it into Explicit and Implicit header routing. In explicit header routing, path information is provided in the address, e.g. a source route. In implicit header routing, other fields in the header influences the routing, e.g. QoS fields like DiffServ are taken into account in addition to the address when the table routing is choosing the final path. He then goes on to divide Explicit header routing into Partial and Complete. Complete is typically source routing. Partial routing is then divided into Hierarchical and Other. The partial hierarchical header routing, where hierarchical addresses are combined with table routing, is the one that scales best and is most commonly used in the global Internet, i.e. IP addresses.

In the paper Paul F. Tsuchiya also argues that what we need are names and routes. Names identify what we want. Routes tell us how to get to a location where we can find an instance of the information object that we want. Locators thus are names that can be used to construct routes. As mentioned he sees two types of routing, header routing and table routing. Locators, or addresses, are an important part of both methods. A locator containing a full source route constitute pure header routing, and then no routing tables are needed. An example of pure table routing would be use of flat names and routing tables holding a routing entry for each name in each router. This would obviously not scale. In practice most routing systems use a combination of header and table routing, i.e. use of names that are to some extent hierarchical, or at least aggregatable, and some type of lookup tables.

An information object can have multiple names. Names can point to other names to provide indirection and aliasing. Names can point to locators, which is what makes id/locator separation possible. So have we made any progress?

We would propose to use the term *Name*, or more specifically *Named Object (NO)*, when talking about a semantic object, i.e. a movie, a web page, a temperature reading, a light switch, a person or a specific network node. An NO is something that you want a copy of, want to talk to or want to manipulate in some way. Here it would be possible to fork off into a semantic web [6] discussion, but we leave that for a future paper.

The other term we propose to use is *Locator*. A locator identifies an instance of a named object and can be used for finding a path to the instance, i.e. route to it. A locator can simply be naming a specific node or attachment point in the network. It does not need to contain any topology or other routing information, e.g. like an Ethernet MAC-address. But it could also contain a full source route. A *Locator* maps to one instance of a *Named Object (NO)*. However the same instance can be reached via multiple locators, e.g. a host that is multihomed can be reached by all locators that its interfaces have.

A locator is also a name, as it is naming a node, an interface or something else that we want to route to. *What makes a certain name a locator is that it exist in a context where it can be used to construct a route.* This is also in line with the traditional layered networking model where what is a name at one layer is an address at another layer as proposed by the OSI model.

IV. So what are the problems?

We will now go on to look at some of the perceived problems with today's Internet architecture from an id/locator split point of view. The problems we discuss below have two things in common (at least). The first is that they represent functionalities that has proven to be difficult to add onto an existing networking architecture. These functionalities also have interdependencies such as if you add them one by one, it will be increasingly difficult to add one more without getting into feature interaction problems. If you e.g. add an end-to-end security mechanism that is designed for point-to-point connections it is unlikely that it will be easy to add point-to-multipoint connectivity without redesigning the security mechanism. The second thing these problems have in common is that it is very common that the proposed solutions for them have id/locator split as a key component.

A. Mobility

Mobility is probably the issue where the semantic overloading of the IP-address is most obviously a key problem. There have been numerous solutions proposed to overcome this where, arguably, IETF Mobile IP and the 3GPP GTP tunneling are the most deployed solutions.

To be able to keep a communication session alive between two communicating entities there is a need to name these entities with names that do not change during the session. These names need to be mapped to locators naming different network attachment points when the communicating entities move relative the networking topology.

The network needs to be able to route packets between network attachment points. There is a need to do this in a globally scalable way. As far as we know this implies routing information needs to be hierarchical, or at least aggregatable in some way. This implies that names of and/or the locators of the attachment points can not change in a way that requires frequent updates of the routing information.

Tunneling solutions like Mobile-IP or GTP tunneling seem unattractive for future networks where we expect the majority of network entities to be mobile, e.g. in a future 5G architecture where mobile and fixed networks are envisioned to converge.

Network mobility where large numbers of hosts change their attachment points at the same time can cause update storms in the routing infrastructure.

B. Naming and address space

Traditionally the need to introduce of a new global naming layer has been regarded as a significant hurdle for deploying a new networking architecture. The slow uptake of IPv6 is a recent example of this. Another key issue here is whether there is a need to introduce a new naming authority for name delegation.

The problem with IPv4 address depletion was one of the early drivers for id/locator separation. This was then largely addressed by the introduction of NATs, at the price of loss of end-to-end functionality. This lost end-to-end functionality has in turn become a new reason to introduce identifiers that the applications can use that are separated from the ephemeral network endpoint identifiers.

Today applications to a large extent use URLs, email addresses or cryptographic hashes as identifiers. These work fine as global identifiers without the need to introduce new naming authorities.

C. Point-to-multipoint and Flash crowds

It was long believed that multicast would provide an efficient solution to point-to-multipoint connectivity in networks. History has proven differently. Today we mainly see overlay solution such as CDNs and peer-to-peer solutions in the global network. Many things that we know well how to do for point-to-point connections still are problematic for multicast. They include: flow control, routing, support for time-shift and how to deal with a storm of subscription requests in flash crowds or other highly dynamic scenarios. As the global Internet today is a unicast network the issue of flash crowds needs to be dealt with using overlay solutions that involve sophisticated provisioning and load-balancing to avoid overloading the network. For true, unexpected, flash crowds there is no solution in today's Internet.

D. Quality of Service (QoS)

Significant effort has gone into introducing QoS support in the Internet, e.g. to be able to deliver realtime services. To do QoS separation, policy based routing is a key component that needs some type of support in naming and addressing. Either

by directly using separate locators for different QoS classes or indirectly by the use of separate header fields, like in DiffServ.

E. Security

The security model in today's Internet builds on that we trust certain servers. We then establish a secure tunnel to the server and trust whatever comes through that tunnel. One problem with this approach is that we are vulnerable to servers being hacked and content being poisoned. This problem is quickly growing as more and more content and services are moved into cloud environments.

The current trend of moving to end-to-end encryption does not solve this problem while it severely limits the possibilities for using network support to enhance services and end-user experiences, e.g. support for multipoint connectivity, caching, network transcoding and compression.

V. Approaches

We will now discuss a number of approaches suggested for accomplishing id/locator separation and ICN. We will point out what we believe was the primary problem(s) that the approach suggests to address and discuss how well it does it. We will also point out any key characteristics of the approach, which sometimes might be a strength that we would like to see in a future network architecture, or it can be a weakness that makes this approach a less attractive candidate for a future network.

1) Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture – 1999: In 1999 J. Noel Chiappa introduces the idea of end-point identifiers [7], primarily to address the host mobility problem. However he also discusses other problems that the multiple use of the IP address brings, including the lack of fate sharing for applications and connections.

2) TRIAD: An Architecture for Content Routing Support in the Internet - USENIX 2000: TRIAD [4] was one of the first projects that introduced the concept of content routing. Their primary concern was the issue of slow and unreliable DNS services that slowed down and made access to content unreliable in the Internet of those days. Something they saw would escalate with higher bandwidths.

3) Internet Indirection Infrastructure (i^3) – SIGCOMM 2002: In the paper [8] the authors generalize the Internet's point-to-point communication abstraction to provide services like multicast, anycast, and mobility. To do that the authors "propose an overlay-based Internet Indirection Infrastructure (i^3) that offers a rendezvous-based communication abstraction. Instead of explicitly sending a packet to a destination, each packet is associated with an identifier; this identifier is then used by the receiver to obtain delivery of the packet." This is one of the first receiver driven communication paradigms.

4) FARA: Reorganizing the Addressing Architecture – SIGCOMM 2003: FARA [9] is one of the first proposals where the authors introduce the idea of mobility for other things than host. The authors have the concept of "entity" as the smallest mobile object that needs to identified. FARA was designed for point-to-point connectivity and proved difficult to extent to the point-to-multipoint case. In a footnote the authors note:

> The FARA model is currently defined only for point-to- point communication between pairs of entities. Initial efforts to extend FARA in a completely natural way to multipoint delivery have been less than satisfactory, which could be considered to be a defect of the FARA model.

5) Host Identity Protocol (HIP) – IETF 2003: Host Identity Protocol (HIP) Architecture, RFC4423 [10] is primarily aiming to make the Internet more secure. It is closely related to the IPsec [11] work. For IPsec to work well two key components are to make sure that you communicate with the right endpoint and that that endpoint remains stable during the communication session. For host mobility the problem of changing IP addresses, when changing network attachment points, need to be dealt with. The introduction of host identifiers addressed both of these problems. By using private/public key pairs to construct the Host Identifier (HI) one could secure the identities of the endpoints. By introducing the HI in-between the application and the IP address a stable endpoint for the IPsec session could be provided. HIP primarily addressed the point-to-point use case.

6) Host Identity Indirection Infrastructure (Hi^3) – SNCNW 2004: Hi^3 [12] combined i^3 and HIP to get secure communication also for point to multipoint communication. It also combined i^3's advantage with receiver initiated communication to mitigate DoS attacks with HIP security.

7) A node identity internetworking architecture (NodeID) – GIS 2006: NodeID [13] is a proposal for how to bridge heterogeneous addressing domains, e.g. IPv4 and IPv6 by introducing a new set of node identities, NodeIDs that are handled at special NodeID gateway routers that are placed between a globally routable core domain (e.g. today's IPv4 Internet) and edge domains using alternative addressing domains. NodeID also propose how mobility, multihoming and security can be supported in this architecture. The use of hashes of public keys for the identifiers is inspired from HIP.

8) Routing on Flat Labels (ROFL) – SIGCOMM 2006: ROFL [14] is proposing to get rid of locators altogether. It introduces the idea do pure name based routing on flat labels. It is inspired by the work on compact routing [15] . While ROFL is routing on names it still uses names to identify nodes. It does not discuss how to name the information objects themselves.

9) Content Centric Networking (CCN) – Google Tech Talks 2006: Content-centric networking (CCN) or Named Data Networking (NDN)[3] was introduced by Van Jacobson when he was at PARC and in August 2006 gave a Google Tech Talk [16]. Many regard this talk as the starting point for what now has grown into the research field of Information-centric Networking (ICN).

CCN [17] has strong similarities with i^3 in the respect that it routes requests for NOs towards the publisher while it checks in for copies on the path towards the publisher.

CCN is similar to IP in the respect that it uses the names of the information objects as locators for routing. CCN is thereby not providing proper id/locator separation. An example

[3]CCN also goes under the name Named Data Networking (NDN), but in this paper we will just refer to it as CCN..

of the problems that this leads to is that it cannot find copies of the requested object in off-path caches. It will also not easily recognize if it in a cache has multiple copies of the same object if they have been published under different names (makes e.g. de-duplication in caches difficult). Furthermore, to support publisher mobility some indirection needs to be added to CCN, otherwise the aggregation of publisher prefixes will not work when some of the aggregated published objects move to a different part of the network topology.

10) Dynamic internetworking based on late locator construction (LLC) – GIS 2007: LLC [18] is proposing how locators can be dynamically constructed based on the current network topology. A key feature of this is that can deal with very dynamic network environments, including mobile networks with very few updates in the routing system.

11) A Data-Oriented (and Beyond) Network Architecture (DONA) – SIGCOMM 2007: DONA [19] like most of the previous approaches addresses security, mobility and multipoint connectivity. In addition DONA is one of the first approaches to bring up the issue of name persistence. DONA introduces names of the format P:L, where P the Principal is a hash of a public key and L the Label is assigned by the publisher that holds the key that P represents. For static content L is expected to be the hash of the content, which offers two features. The first is that it gives the name self-certifying properties. The second is that it gives persistence to the name, something DONA is one of the early promoters of.

The names that do not hold any topological information are registered in Resolution Handlers (RH). The RH are domain specific, e.g. to publisher domains. To resolve the names into routable transport locators resolution requests are routed by name to the RH that can resolve it. The use of RHs makes it possible for DONA to also support publisher mobility. DONA builds on a number of previous approaches including TRIAD and HIP. It also borrows ideas for session initiation from SIP. It can be noted that the scalability of the name-based routing that DONA proposes can be questioned.

12) Network of Information (NetInf) – ReArch 2008: The initial NetInf ideas are presented in the paper *Design considerations for a network of information* [20] is primarily combing ideas from DONA and CCN, but also i^3 and HIP have had mayor impact on the NetInf architecture. A key design criterion for NetInf was to have good support for mobility, including publisher mobility. This lead to the choice of names without topological significance but that can support name-data integrity by using the hash of the content in the name. The use of a Name Resolution Service (NRS) provides the indirection needed for publisher mobility.

The original NetInf design separates between Information Objects (IO), Data Objects (DO) and Bit-level Objects (BO). The IO is the semantic object, e.g. a certain musical song. The DO is a particular instance of the IO, e.g. a specific MP3 file. The BO is an actual copy of the DO residing on e.g. a hard disk in a node.

13) Publish Subscribe Internet Routing Paradigm (PSIRP) - ICT-MobileSummit 2008: In the paper *RTFM: Publish/Subscribe Internetworking Architecture*[21] PSIRP [4] is presented as a true clean slate architecture. It uses a number of identifiers. Application (Level) Identifiers (AId) that is their name for a semantic object. Rendezvous Identifiers (RId) that allow subscribers and publishers to meet within the rendezvous system. Forwarding Identifiers (FId) are Bloom filters that constitute a source route and are placed in the packet header. This makes PSIRP a pure header routing technique as no forwarding state is needed to be kept in PSIRP routers. In addition PSIRP also uses additional identifiers: Scope Identifiers (SId) and Algorithmic Identifiers (AlgIDs).

14) Locator/ID Separation Protocol (LISP) - 2008: The primary motivation for LISP was to try to limit the growth of the routing tables in routers. This problem was discussed at the IAB's October 2006 Routing and Addressing Workshop (RFC 4984). LISP [22] enables separation of IP addresses into two new numbering spaces: Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). No changes are required to either host protocol stacks or to the "core" of the Internet infrastructure. LISP is primarily a way to manage the IP address space. The EID only refers to hosts, not to named objects.

VI. ICN Outlook

We will now look at which are the remaining main issues that need to be resolved for ICN to finally provide the long desired id/locator separation. Security is, as always, a key issue for any future networking architecture. Instead of listing it as a general issue we have chosen a few security issues that relate to the id/locator separation, see the first three issues below.

A. Name persistence

Name persistence is an issue the importance of which will grow over time. Already today broken URLs is a problem, as we experienced in writing this paper. In a future ICN network where the named objects are the primary architectural components it is critical that their names remain constant and are independent of who has published them or where they are located. A way to achieve this is to name immutable objects by the hash of the object. The RFC 6920 [23] "Naming things with hashes" propose how this can be done. A remaining challenge is how to provide a lookup service for such objects that is globally scalable.

B. Access control

In ICN objects are expected to reside in untrusted caches including end-user devices. Considering the problems keeping good access control in cloud environments, which are managed environments, this is a major challenge. Encryption is one possibility. Some issues to address here is how to encrypt in a way that we don't lose the cacheability of the objects and/or get into severe key distribution problems. A promising technology here is Attribute Based Encryption (ABE) [24] .

C. Anonymity and accountability

ICN technology can be used to build a network that provides perfect anonymity or that are truly Orvellian. How to balance anonymity and accountability is primarily a regulatory issue, not so much a technology issue. Also the market forces that want to count your clicks and in all ways map your habits and preferences should not be underestimated.

[4] PSIRP is sometimes referred to by the name PURSUIT, but we will just use PSIRP in this paper.

D. Publisher mobility

Mobility was one of the very early motivations for id/locator split and remains one of the most obvious reasons why it is needed. It is crucial that any future networking architecture supports mobility, including publisher mobility, inherently and that mobility will not be an add-on.

E. Deployment

For it to be feasible to deploy an ICN approach it must be possible to run it as an overlay on the existing IP infrastructure. Preferably it should help in the introduction of IPv6 by bridging between the two IP domains. However, ICN should not rely on IP but be able to run on top of any type of bit transport. The most apparent application domain for ICN is media distribution as it integrates into the network CDN and peer-to-peer functionality that today is provided as overlays. In particular support for coping with flash crowds for live video streams can be a driver as there is no good support for that in the current Internet. Another interesting application domain is IoT. For IoT a new networking environment that supports DTN is needed. One proposal for this is CoAP, which already supports ICN functionality such as a lookup service in the proposed Resource Directory.

VII. Conclusions

We expect the future Internet architecture to be a much more information-centric networking architecture compared to the host-centric networking architecture that we have today. It will also have inherent support for mobility, multihoming, security, etc. We also believe that the networking API will change from today's point-to-point oriented socket API to an information object oriented API where the application asks the network for an information object by name without having any idea of which box in the network that delivers it. Such an API will look very much like a database API. If this vision comes true we believe that the id/locator issue will finally disappear, as it is inherent in this type of content networks.

Acknowledgments

We like to express our thanks to Anders Eriksson and Bengt Ahlgren for providing valuable comments on this paper.

References

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine - Speical Issue on Information-Centric Networking*, July 2012. (to be published).

[2] J. F. Shoch, "Inter-network naming, addressing, and routing," in *Proceedings of the Seventeenth IEEE Conference on Computer Communication Networks*, (Washington, D.C.), pp. 72–79, IEEE, 1978.

[3] P. F. Tsuchiya, "Efficient and robust policy routing using multiple hierarchical addresses," *SIGCOMM Comput. Commun. Rev.*, vol. 21, no. 4, pp. 53–65, 1991.

[4] D. R. Cheriton and M. Gritter, "TRIAD: A new next-generation Internet architecture." http://www-dsg.stanford.edu/triad/, July 2000.

[5] J. H. Saltzer, "On the naming and binding of network destinations," in *Proceedings IFIP/TC6 International Symposium on Local Computer Networks* (P. R. et al., ed.), (Amsterdam, Holland), pp. 311–317, North-Holland Publishing Company, April 1982.

[6] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web - a new form of web content that is meaningful to computers will unleash a revolution of new possibilities," *Scientific American Special Online Issue*, vol. 4, pp. 24–30, 2002.

[7] J. Chiappa, "Endpoints and endpoint names: A proposed enhancement to the internet architecture," *Private Communication*, 1999.

[8] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *Proc. ACM SIGCOMM*, (US), August 2002.

[9] D. Clark, R. Braden, A. Falk, and V. Pingali, "Fara: Reorganizing the addressing architecture," *ACM SIGCOMM Computer Communication Review*, pp. 313–321, 2003.

[10] R. Moskowitz and P. Nikander, "Host identity protocol architecture." RFC 4423, jan 2004.

[11] S. Kent, "Rfc 4303," *IP Encapsulating Security Payload (ESP)*, 2005.

[12] P. Nikander, J. Arkko, and B. Ohlman, "Host identity indirection infrastructure (hi3)," in *Proc. Second Swedish National Computer Networking Workshop (SNCNW), Karlstad, Sweden*, 2004.

[13] B. Ahlgren, J. Arkko, L. Eggert, and J. Rajahalme, "A node identity internetworking architecture," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–6, IEEE, 2006.

[14] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, and I. Stoica, "ROFL: Routing on flat labels," in *Proc. Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '06)*, (New York, NY, USA), pp. 363–374, ACM Press, 2006.

[15] M. Thorup and U. Zwick, "Compact routing schemes," in *Proceedings of the Thirteenth Annual ACM Symposium on Parallel Algorithms and Architectures*, SPAA '01, (New York, NY, USA), pp. 1–10, ACM, 2001.

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

[17] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th ACM Conf. Emerging Networking EXperiments and Technologies (ACM CoNEXT)*, (Rome, Italy), December 2009.

[18] A. Eriksson and B. Ohlman, "Dynamic internetworking based on late locator construction," in *IEEE Global Internet Symposium, 2007*, pp. 67–72, IEEE, 2007.

[19] 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 *Proc. Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '07)*, (New York, NY, USA), pp. 181–192, ACM Press, 2007.

[20] B. Ahlgren, M. D'Ambrosio, M. Marchisio, I. Marsh, C. Dannewitz, B. Ohlman, K. Pentikousis, O. Strandberg, R. Rembarz, and V. Vercellone, "Design considerations for a network of information," in *Proceedings of the 2008 ACM CoNEXT Conference*, p. 66, ACM, 2008.

[21] M. Särelä, T. Rinta-aho, and S. Tarkoma, "Rtfm: Publish/subscribe internetworking architecture," in *Proceedings of the ICT-Mobile Summit 2008* (P. Cunningham and M. Cunningham, eds.), 2008.

[22] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "Rfc 6830: The locator/id separation protocol (lisp)," 2013.

[23] S. Farrell, D. Kutscher, C. Dannewitz, B. Ohlman, A. Keranen, and P. Hallam-Baker, "Naming Things with Hashes." RFC 6920 (Proposed Standard), Apr. 2013.

[24] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, ACM, 2006.