CLIP: Content Labeling in IPv6, a Layer 3 Protocol for Information Centric Networking

Laura Heath, Henry Owen, Raheem Beyah
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250
lheath3, owen, rbeyah@gatech.edu

Radu State
Interdisciplinary Centre for Security,
Reliability and Trust
University of Luxembourg, L-1359 Luxembourg
radu.state@uni.lu

***Abstract*—A great deal of research has been done in the last several years on information centric networking (ICN), where named data items, rather than end host identities, are the primary routable entities. Several prototypes have been proposed, but all face daunting concerns in the areas of performance, complexity, backwards compatibility, and user security and privacy. We propose a protocol which creates globally-unique data item names and embeds these names, plus associated metadata, into an IPv6 header. We then show how this allows the use of the IPsec suite of protocols to mitigate user privacy and security concerns. Next, we show that using RFC-compliant IPv6 datagrams as the named content allows content routing to be done using standard gateway protocols and also ensures backwards compatibility with unmodified networks. Lastly, we give an example of how the CLIP header's structure will simplify the design of automatic caches in the network.**

***Index Terms*—Information Centric Networking, IPv6, content distribution**

I. Introduction

There is a growing realization that some early design decisions while developing the Internet have proven to be significant constraints to the performance and scaling of the network today. In particular, there was an unspoken assumption that all communication was host to host, and each host was one unique computer at one fixed location. Yet today, the majority of the data transfers over the Internet are either data distribution or provision of services—things that are not inherently (or even preferably) single-host to single-host point-to-point. A great deal of network research and development over the last three decades has been devoted to finding ways to make a point-to-point design function in a non-point-to-point way—such as multicasting (one to many transmission), proxies (inserting an intermediary between two hosts in the communications stream), NAT translation (multiple hosts sharing one routable IP address), mirror sites (duplicate hosts in physically separate locations to serve identical content), and more. Each of these configurations have produced a crop of complicated workarounds, which frequently break other services unexpectedly and lead to yet another *ad hoc* patch.

In recent years, there have been multiple projects focused on addressing the root cause of this complexity: what users want is specific data, but what the network provides is connection to a specific host. The new network designs seek to meet the need for named data by elevating content item names to first-class, routable network entities. This paradigm is called *information centric networking* (ICN) [1], and several distinct prototypes have been developed.

All of the proposed prototypes face several critical questions. The most important is, what will the network's "narrow waist" (the lowest-level global means of establishing communication)—if it is IP, as in the current Internet, how does the ICN system interact with IP, and how is it any different than an overlay system (such as a peer-to-peer or content delivery network) used today? And if the narrow waist becomes a name-based global network primitive, how can the system guarantee the same speed and resilience as IP, if it is working on a much larger set of much less-structured names?

Two additional hard questions pertaining to the network's routing primitives follow: first, what is the size of the named data items? The size of the data items drives the growth of the routing tables needed, which is expected to be a significant performance limit for these systems. Second, what are the policies and procedures necessary for internetworking, *i.e.*, effectively and safely routing ICN data through networks that are not under your control, and have divergent equipment and policies?

Lastly is the question of network security, especially end user privacy. If data is routed by name, then it would seem that any ICN system would necessarily expose the name of all data accessed by end users. In contrast, the legacy network permits end users to securely establish an encrypted tunnel between hosts, so that eavesdroppers will only know that a quantity of data has been transmitted, but will have no knowledge of the content.

A. Overview of our approach.

Our approach begins with embracing IP as the narrow waist for ICN: it is arguably the most successful design in the history of human engineering, and it is effectively ubiquitous in networks worldwide. Our contribution is twofold: first, we describe a structured way to form a globally-unique name for data items. Next, we exploit the larger size and more generic structure of the IPv6 header to embed the content item name and associated metadata into the header in a way that ensures the datagram is RFC-compliant. This becomes the named data item for the ICN network.

Once this is done, we show that many of the serious design concerns with existing ICN networks can be solved with current layer 3 protocols. First, we discuss how the IPsec suite of protocols can be used both to authenticate individual data items and to provide privacy and security for end users. Next, we show that, by accepting IPv6 as the narrow waist, we can depend on well-studied, existing routing protocols to ensure global reachability, network stability, and performance. Third, we show that the wide latitude in datagram size permitted under IPv6 gives content owners significant flexibility to size their content items appropriately to their operational requirements, without a concomitant runaway growth in routing table size. Lastly, we give an example of how CLIP could be used to simplify the automatic data caches that most ICN systems rely on.

B. *Related Work.*

There has been a significant amount of recent research focused on ICN. One of the first attempts at content centric networking was the Data-Oriented Network Architecture (DONA) [2], which built on the work of TRIAD [3]. The principal features of this system were 1) creation of a self-certifying (cryptography-based) naming convention for individual data items, 2) replacing domain name server (DNS) architecture with a name-based anycast to map these names to an IP address, and 3) pervasive caching at each node to improve availability.

Subsequently, the National Science Foundation's Future Internet Architecture program selected the Named Data Networking [4] project for funding; the system has further developed into Content Centric Networking (CCN) [5, 6]. In this system, names for data items are hierarchical and human readable (much like URLs), and routing is done by a longest-prefix match against the name. That is, this system takes the CCN name-based routing as the narrow waist, and uses IP (or some other protocol) as a transport means only. Routing is done by having each node maintain a list of pending requests and cache of previously satisfied requests. Nodes propagate unfilled requests to their neighbors, which in turn will either reply with the data needed or log a pending request and forward the request to their own neighbors.

A third major ICN proposal is PURSUIT project [7], which follows on from the Publish-Subscribe Internet Routing Paradigm (PSIRP) project [8]. Content items are named with a statistically unique, self-generated flat label, which are published into a named, hierarchical "scope" of related information. Three logically separate (though possibly physically combined) nodes are used: a rendezvous node matches requests and publications, a topology manager determines the path, and a forwarding node prepends a series of Bloom filters to the data. Nodes in the forwarding fabric do not have forwarding tables; rather, they simply apply the Bloom filter to choose the correct interface to use [9].

Note that both CCN and PURSUIT opt for replacing the narrow waist of IP with a new name-based primitive—but that there is no equivalent of the Gao-Rexford stability and reachability conditions for these algorithms. Also, note that each has *de facto* developed a naming convention which concatenates the identity of the data publisher (in CCN, the top level name; in DONA, the owner of a specific public key; in PSIRP, the owner of a scope) with a unique label for the content item itself. Our naming convention follows the same general concept, but embeds each in a different field in the IPv6 header.

II. Partitioning of the IPv6 Address Space for Content Labeling

The first step in our process was to create a globally-unique name for each content item. As noted above, we adopted a naming convention of associating a globally-unique publisher label (PL) with a unique content item label (CL). We then integrated this naming convention into the IPv6 address space. Our approach is to create a large subnet for all ICN traffic, then create individual subnets for each content publisher, and lastly to append the content label to the interface ID and destination options header.

We define three types of PLs: global PLs, which are unique to a specific organization or user; local PLs, which are non-routable and managed by the local network provider; and anonymous PLs, which are designed for ephemeral use.

A. *Global Publisher Labels (PL)*

The global publisher label (PL) will have to be assigned by a universally-recognized organization, similar to the ways that IP addresses, MAC addresses, and AS numbers are issued today. While this is a less-than-ideal requirement, in practice there is no other way to ensure that the publisher label is actually unique. Indeed, as the PL will become part of the globally-routed IP address, it may make most sense to have these numbers issued in exactly the same way that IP addresses are allocated today. Additionally, this partially resolves the problem of associating a network-level item (the PL) with a real-world identity of a person or organization (this is discussed further in section III).

In our work, we assume that the length of PL is 64 bytes long, and begins with 0xC. The length is chosen because the smallest allocation of IPv6 address space to be deployed is the /64 subnet, as described in RFC 6177 [10] and RFC 4291 [11]. Thus, we may be certain that there is at least 64 bytes of address space available for assignment on any RFC-compliant network. The prefix 0xC is needed for two reasons. First, a unique subnet prefix for all ICN content (from all publishers) makes subsequent special handling by network nodes much easier to implement in hardware. A unique subnet prefix allows aggregation of the ICN address space and separates it logically from the legacy point-to-point IP space, easing the administrative overhead of policy enforcement. Second, this prefix is not currently specified for use in any other network addressing schemes, including stateless autoconfiguration [12] and IPv4 embedded/compatible/mapped addresses [13]. This gives a global space of 2^{60} unique names for publishers, while still leaving a very large number of subnets available in each network address for legacy host names.

B. *Local Publisher Labels*

On the link local address FE80::/10, the publisher label may be locally assigned by the network provider. This will allow for locally produced and consumed data to be locally administered as well. We expect that most users, and possibly even individual processes on local hosts, will receive ID

numbers of this type for intra-network use. The network provider may make arrangements to republish local data under a commercial or organizational publisher label for any data needing global availability.

Readers may question why we are not using the unique local addresses (FC00::/7) defined in RFC 4193 [14] for this purpose. There are two main reasons. First, the RFC specifically requires the remaining 40 bits of the network address to be generated randomly and further specifies that these addresses must not be aggregated by routers. In contrast, under this protocol, network providers would be expected to structure and aggregate the network addressing scheme to simplify administration and logical data flow. Second, the early experience with "site local" addressing showed that addresses can "leak" via the application layer, particularly with mobile or multi-homed hosts [15]. This can lead to erroneously addressed packets flooding through the network. (For example, a user would download his e-mail at work using one addressing scheme, but attempting to use the same address from home would cause an IP number conflict on his ISP's network.) By using the link-local addresses, this problem is largely avoided: the network does not route FE80::/10 addresses, so an IP address conflict could only occur if another host on the same layer-2 switched link had both the same global ID, local ID and local publisher ID (120 address bits total).

C. Ephemeral or Anonymous Publisher Labels

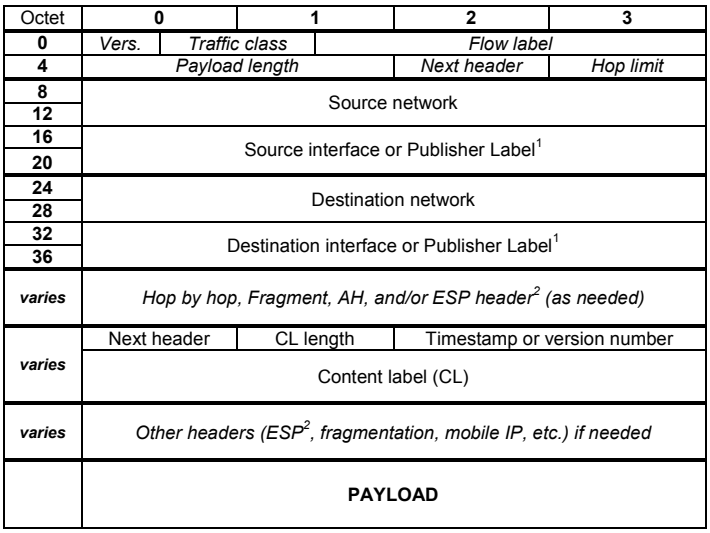
Lastly, to address the need for no-cost and anonymous publisher labels, we reserve the PL numbers beginning with 0xCFF as a type of "unlicensed spectrum" for anonymous or ephemeral use. We foresee two main uses for addresses in this class. First, we expect that there will be a need for ephemeral "rendezvous" points, particularly for one-time data transfers between individuals. Second, these addresses can be used to conceal the true publisher providing data to a host, as described in section III. B., below.

D. Content Item Label (CL)

The last component needed for this system is a unique content item label. The label begins with a two-byte field for timestamp or version number, assigned by the publisher. Next, the publisher must assign a unique content item label. The question of content naming is an area of dispute among the designers of different ICN systems, with some systems choosing a human-readable, hierarchical naming convention (similar to the URLs used in HTTP), while others use a cryptographically-derived self-certifying name. This protocol functions equally well with either convention. (But see section III.A., where we describe why we believe that cryptographic-based names are not desirable in practice, and how to provide the necessary binding using IPsec protocols.)

The CL is placed into the destinations options header field in the IPv6 datagram as shown in Fig.1. This has three main advantages: first, the content label can be very large—up to the size of the datagram (a minimum of 1280 bytes on any network and a maximum of 2^{32}-1 bytes for a jumbogram). Second, this allows the label length to be variable—the only system requirement is for the header to be a multiple of eight bytes long.

Third, the IPv6 specification states that the destinations options header is not read or acted on during transit. This means that the legacy network will not need any additional equipment or policy to handle ICN traffic. A deployment of the ICN system does not require any arrangements with service providers beyond receiving IPv6 service. However, if a network provider chose to implement an information-centric proxy or cache, the content information would be easily available in a standard structure (discussed at further length in section IV.C.). That is, they could implement this proxy or cache without the need for deep packet inspection, statistical tests on variable length windows, etc.—and in fact, without standing up a full ICN network.



1. When a host is requesting a content item, the destination interface will be the Publisher Label (PL); when a content server is replying with a content item, the source interface will be the PL

2. The ESP header may appear before or after the Content Label (CL); see section III.B for discussion.

Fig. 1. Format of the datagram header

III. Security Considerations

There are three main security considerations that any deployable ICN system must address: key management; the secure binding of real-world identity of the publisher, publisher label, content label, and the content itself; and preserving the privacy of end users. As noted above, we assume that the binding between real-world identity and PL is done when the PL is issued. An addressing and header protocol such as this one also cannot address the question of key distribution. We assume that this is handled by another method, such as DNS IPSECKEY records [16] or Kerberos [17], and that a security association (SA) has been created.

A. Secure binding of publisher label, content label, and data

Once a key has been distributed , the binding of the PL, CL, and data can proceed. IPv6 provides a straightforward solution with the Authentication Header (AH) feature of the IPsec suite of protocols [18]. It binds all non-mutable elements of the datagram (*i.e.,* everything except the traffic class, flow label, and hop limit) with a cryptographic signature. The signature may be based on either symmetric or public keys, and the protocol allows for new cryptographic algorithms to be implemented in the future. We believe this flexibility is critical

for a long-term production system, to allow for security and performance upgrades where and when needed.

Note, however, that there are some important conceptual differences between the legacy point-to-point security association and an ICN security association. Most importantly, the use of challenge-and-response methods (like encrypting nonces) and mutually-constructed secrets (as in the Diffie-Helman algorithm) is not generally possible, as there is no explicit connection between the publisher and receiver. Additionally, in the context of this protocol, a "session" is not an end-to-end connection between two hosts, but rather a period of authorized access to certain CLs.

It is also important to discuss the question of using cryptographically-generated, or *self-certifying*, names as a security feature to prevent denial of service attacks. As described in [19], under this concept publishers are identified by a fixed-length hash (P) of their public key (assumed to be bound to their real-world identity by a DNS-type lookup system). The data item is named by concatenating P with a label L, which is in turn generated by cryptographically hashing the tuple <data, public key, L, metadata, signature>. This tuple is the data item provided to a requestor. This makes it effectively impossible to impersonate a publisher, as any node can hash the provided public key and verify both P and L.

We believe this is extremely risky in a production network, however. Key compromises are an unfortunate reality, and when (not if) they occur, every data item that the victim has ever created must be renamed—and the new names will have no relation to the compromised names. Such an event would be incredibly traumatic for the victim, particularly if the compromised network cannot be shut down for overhaul. Even in the best case, cryptographic algorithms have a finite lifespan, and production environments typically require a transition period measured in years; during that time, both algorithms must be supported. If names are generated by specific hash algorithms, an overlap would seem to require either duplicate copies of every data element (one using the old hash, and one using the new) or some type of name translation system for every data element. In light of these difficulties, we do not recommend using the naming scheme to implement security features.

B. *Preserving the privacy of the end user*

The second issue to address is the issue of privacy for the end user. ICN systems route and cache datagrams based on the names of the publisher and content item; as a result, anyone can determine what data the end user is accessing. This could have serious negative consequences for the end user. We believe that our new protocol, combined with the IPsec suite of protocols, can effectively mitigate some of these issues.

Using the AH header alone, as described above, does not provide privacy services will leave both the identity of the publisher and the content of the data item easily available to any observer. This has the same privacy issues currently experienced with unencrypted internet traffic. The remedy for this is to apply the Encapsulating Security Payload (ESP) protocol [20] in one of three configurations.

The first privacy option is to apply ESP in transport mode to encrypt the payload only. In transport mode, only data that comes after the ESP header is encrypted, so in this case, the ESP header is placed after the CL and before any transport-layer headers (including port and protocol numbers). The AH must also be used here to bind the PL and CL to the payload. Observers will know the PL, CL, and the approximate size of the payload, but not the actual content. (As an aside, publishers are free to generate CLs for their data items which might appear to be gibberish to an outsider.)

The second option is to move the ESP header in front of the CL. In this case, ESP will conceal all information about the CL (even its length). An observer will know that the end host received an (approximate) amount data from a particular publisher, but no other details are visible. This will, however, block the network's ability to cache the data by name.

Lastly, in cases where the user wishes to conceal the PL, CL and payload contents, the publisher would use ESP in tunnel mode. In this mode, the entire original packet is encrypted, and a new header generated. This new header would have either a republisher's PL or an ephemeral PL (as described in section II.C.), and an ephemeral CL. This will conceal the publisher, content name, and payload. This provides the full spectrum of layer-3 security and privacy features available to legacy network users.

IV. Performance Considerations

As noted in the introduction, there are also several critical performance issues that plague ICN designs. We believe that use of our new protocol has the potential to reduce or eliminate problems in the following areas: interdomain routing, object sizing, and automatic caching.

A. *Interdomain Routing and Policies.*

There are two separate problems to address in interdomain routing for ICN networks: first, the network needs to demonstrate that it is *safe* (guaranteed to converge), *autonomous* (does not require coordination or exposure of intra-network data), and *expressive* (allows for diverse policies between autonomous systems). Within a single network, one might assume that the network administrator is able to observe and rectify any routing abnormalities (such as unreachable addresses, loops, and route flapping). However, no such guarantee is available for internetworking, which thus requires much more rigorous assurance from the protocols themselves. Second, an ICN network needs to address the issue of *backwards compatibility* with the legacy Internet—whether hosts not directly connected to an ICN network can access ICN content.

A general answer to the first question is elusive for current ICN prototypes. As noted in Section I., each ICN system handles routing in a different way. From a theoretical standpoint, conditions for global reachability and stability in name-based routing have not been determined. From a practical standpoint, little actual data is available on the performance and safety of these designs, particularly concerning the effects of divergent policies on internetwork routing. In contrast, by using IP natively for naming and routing, we are able to apply the extensive theoretical and practical results of working with BGP directly to the problem. ICN-specific routing policy issues could be addressed using the BGP Extended Communities Attribute [21, 22]; note also that there is extensive ongoing research in methods of passing

routing and link state information from both interior and exterior gateway protocols up to the application layer that may be useful for ICN-enabled devices.

The question of backwards compatibility has only been addressed in the sense of ensuring that a given ICN network can traverse an IP-based network. That is, the ICN is either an overlay on top of the IP transport network, or it tunnels packets from one ICN "island" to another across an IP network. (In Section II, we discussed the fact that CLIP datagrams can transit legacy networks without modification.) Backwards compatibility in the full sense, that is, whether hosts on an unmodified legacy network can access ICN content and hosts on an ICN network can access data on the legacy network, is not possible when the ICN network does not use the IP waist; non-ICN routers will not recognize or route ICN data.

CLIP solves part of the problem. Namely, any Internet-connected host can natively reach any ICN-enabled device containing a specific named data item. The remaining missing pieces are 1) an ICN equivalent of TCP, to handle transport-layer functions for the host applications, 2) a content advertisement and discovery protocol (all of the ICN systems under discussion have some variant of the publish/subscribe model; similar architectures are also common in many Web applications [23]), and 3) some method of secure key distribution and possibly authentication (as noted in Section III, above). Note that all of these must be solved for the ICN network itself to function as well.

B. *Object sizing.*

Another area of strong disagreement between ICN proposals is the size of the individual objects. The principal concern is that, by naming data objects instead of end hosts, the number of named items to be located by the network expands by many orders of magnitude. The smaller the size of the named data item, the worse the scaling problem becomes. Recent results indicate that it is physically impossible to route ICN-named packets at line speed and Internet scale using current memory technology [19, 24].

In one sense, our new protocol bypasses the problem: it is pure IPv6, and so by definition is routed at line speed (*i.e.,* as fast as legacy traffic is now). The content-specific portion of the name is placed in the destinations options header and can be ignored in transit (but see our discussion about automatic caching below). Our definition of a data item is a datagram, but in IPv6 the size may notionally be anywhere from one byte to 4 GB. While the extremes are unlikely to have any real-world application, this does demonstrate that content owners will have wide latitude to size their objects appropriately for their operational needs under this naming convention. For example, one would expect that bulk transfer objects could be quite large, while streaming media content would likely be broken into very small, sequentially accessed objects. Likewise, we expect publishers to choose CLs for their data which group similar content together, to improve the storage and retrieval performance of their data.

This proposal produces a system where there is a relatively long name-to-address resolution process (handled above layer 3), followed by line-speed routing along the hop-by-hop best path, as determined by current network hardware, software and policies. The IP addresses are designed to be aggregated as in the legacy network, and so routing table scaling should not be worse than already incurred by the legacy network traffic.

C. *Automatic Caching.*

Another key performance feature of the ICN systems under discussion is extensive caching. Typically, every data item is expected to be cached (including items simply transiting the network) and every network node is expected to cache data. Some authors [1] argue in contrast that, while there is significant benefit from moderate-sized caches handling the most popular content, the statistical distribution of the "long tail" of less-popular content is highly unfavorable to caching. While no ICN network is currently handling production traffic, a similar function is being performed by content-aware Web proxies, which use statistical tests on sliding windows to cache pieces of Web pages (including redundant parts of "uncacheable" Web pages). An empirical study [25] in 2010 of their effectiveness concludes that, for general Web traffic, average byte hit rates can be optimized at 42-50%, if the average chunk size is 128 bytes, and cache size is on the order of four times total network traffic. These numbers are encouraging for ICN networks.

Regardless of whether the network is configured to cache at every node, or only at one or more appliances on the network, we believe that the structure of the CLIP header will make the caching function easier and faster to implement in hardware. There are two reasons we believe this: first, the content boundaries are determined by the publisher, based on knowledge of the content. This should produce a better result (and will certainly be faster) than statistical estimation. Second, the PL, CL, and timestamp are in a predictable location and known format, which means the required processing is simply an XOR against the header. If the datagram was content-addressed, its PL would be checked against the cache contents; if matched, and the CL and timestamp would be checked next. If the data item were present, it would be served from the cache. If there were no match at any step, or if the cache failed (too busy or broken), the datagram would return to the critical path and be routed normally, *i.e.,* the system would "fail soft". (Routing on a pure ICN basis could fail entirely if the cache router became overloaded.) Writes to the cache would have to be protected by a stateful firewall (only replies to earlier requests accepted) and a cryptographic check of the AH and/or ESP data; however, writes would take place off of the critical path.

V. Future Research

Defining this protocol is only the first step to a feasible full-fledged ICN system. One approach to verify and validate our proposed method would be to integrate CLIP into one of the existing ICN prototypes. While this is possible, it would be difficult to achieve because these prototypes have avoided the IP waist, whereas CLIP is designed to maximize use of it. Another approach to verify and validate our approach would be to design the layer 4 (and above) protocols that would be needed to build out a basic ICN-IP system—an ICN equivalent of TCP, and a publish-subscribe content advertisement system. While our research has an ambitious agenda, it is not by any means uncharted territory; indeed, one might say that content delivery networks, peer-to-peer networks, and content-aware

HTTP proxies are all special-purpose ICNs, and they might well serve as the base for a more general ICN solution also.

REFERENCES

[1] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, and J. Wilcox, "Information-Centric Networking: Seeing the Forest for the Trees," in *Proceedings of the ACM HOTNETS '11*, Cambridge, MA, USA, November 2011.

[2] T. Koponen, *et al*. "A Data-Oriented (and Beyond) Network Architechture." In *SIGCOMM '07*, Kyoto, Japan, Aug. 2007.

[3] `www-dsg.stanford.edu/triad`

[4] `http://www.named-data.net/`

[5] V. Jacobson, *et al*, "Networking Named Content", in CoNEXT '09, Rome, Italy, 2009.

[6] `www.ccnx.org`

[7] `www.fp7-pursuit.eu/PursuitWeb`

[8] `www.psirp.org`

[9] P. Jokela, A. Zahemsky, C. Rothenburg, S. Arianfar, and P. Nikander, "LIPSIN: Line Speed Publish/Subscribe Inter-Networking", in *Proceedings of the ACM SIGCOMM '09*, Barcelona, Spain, August 2009.

[10] T. Narten, G. Huston, and L. Roberts, IPv6 Assignment to End Sites, RFC 6177, IETF, March 2011.

[11] R. Hinden and S. Deering, The IP Version 6 Addressing Architecture, RFC 4291, IETF, February 2006.

[12] S. Thompson, T. Narten, and T. Jinmei, IPv6 Stateless Autoconfiguration, RFC 4862, IETF, September 2007.

[13] C.Bao, C. Huitema, M. Bagnulo, M. Boucadaire, and X. Li, IPv6 Addressing of IPv4/IPv6 Translators, IETF, October 2010.

[14] R. Hinden and B. Haberman, Unique Local Unicast IPv6 Addresses, RFC 4193, IETF, October 2005.

[15] C. Huitema and B. Carpenter, Deprecating Site Local Addresses, RFC 3879, IETF, September 2004.

[16] M. Richardson, "A Method for Storing IPSEC Keying Material in DNS", RFC 4025, IETF, February 2005.

[17] C. Neuman, T. Yu, S. Hartman, K. Raeburn, "The Kerberos Network Authentication System v5", RFC 4120, July 2005.

[18] S. Kent, IP Authentication Header, RFC 4302, IETF, December 2005.

[19] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in Content-Oriented Architectures", in *Proceedings of ACM SIGCOMM ICN '11*, Toronto, Canada, August 2011.

[20] S. Kent, IP Encapsulating Security Payload (ESP), RFC 4303, IETF, December 2005.

[21] Y. Rekhter, IPv6 Address Specific BGP Extended Communities Attribute, RFC 5701, IETF, November 2009.

[22] S. Sangli, D. Tappan, and Y. Rekhter, BGP Extended Communities Attribute, RFC 4360, IETF, February 2006.

[23] 23 P.T. Eugester, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The Many Faces of Publish/Subscribe", *ACM Computing Surveys*, 35(2), June 2003.

[24] D. Perino and M. Varvello. "A Reality Check for Content-Centric Networking." In *Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking, ICN'11,* Toronto, Canada, Aug. 2011.

[25] S. Ihm, V. Pai, "Towards Understanding Modern Web Traffic." Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement (IMC '11), Toronto, Canada, Aug. 2011