SNAMP: Secure Namespace Mapping to Scale NDN Forwarding

Alexander Afanasyev
UCLA
afanasev@cs.ucla.edu

Cheng Yi
University of Arizona
yic@cs.arizona.edu

Lan Wang
University of Memphis
lanwang@memphis.edu

Beichuan Zhang
University of Arizona
bzhang@arizona.edu

Lixia Zhang
UCLA
lixia@cs.ucla.edu

***Abstract*—Named Data Networking (NDN) is a proposed information-centric design for the future Internet architecture, where application names are directly used to route requests for data. This key component of the architecture raises concerns about scalability of the forwarding system in NDN network, i.e., how to keep the routing table sizes under control given unbounded nature of application data namespaces. In this paper we apply a well-known concept of *Map-and-Encap* to provide a simple and secure namespace mapping solution to the scalability problem. More specifically, whenever necessary, application data names can be mapped to a set of globally routable names that are used to retrieve the data. By including such sets in data requests, we are informing (more precisely, hinting) the forwarding system of the whereabouts of the requested data, and such hints can be used when routers do not know from where to retrieve the data using application data names alone. This solution enables NDN forwarding to scale with the Internet's well-understood routing protocols and operational practice, while keeping all the benefits of the new NDN architecture.**

I. Introduction

As Internet applications become increasingly data-centric, a number of new network architecture designs ([1], [2], [3], [4]), including Named Data Networking (NDN [5], [6], [7]), have proposed a data-centric communication paradigm, which in part requires the network forwarding system to deliver packets directly based on data names. In short, data retrieval requests in NDN (interests) carry names of the desired data (e.g., "`/net/ndnsim/www/index.html`"), and routers use these names to forward the interests towards the closest copy of the data. When the data is found, it is returned to the requester using the recorded interest state at the routers. By explicit naming of the data and binding of this name with a cryptographic signature, NDN provides a number of benefits including data-centric security, support for universal in-network caching, built-in multicast delivery, and better alignment between the desired application usages and the underlying data delivery model in general. However, one frequently raised concern about NDN is the scalability of its name-based forwarding ([8], [9]). Given that the number of data names is unbounded, can one keep the size of name-based NDN forwarding information base (FIB) under control?

Forwarding scalability concern is not new. Although IP address space is finite, IP forwarding (routing) scalability has been considered one of the major challenges since the early days of the Internet deployment. One basic approach to keep the global IP routing table under control is address aggregation by allocation: end users and small networks get their addresses from the access providers, and access providers inject only the aggregated prefixes into the global routing system. However, a number of factors, including a growing demand for provider-independent addresses [10], network-layer traffic engineering and load balancing [11], and mitigation of DDoS attacks and prefix hijacks [12], have been driving the growth of the DFZ routing table size. Another solution to keep the routing table size under control is to introduce a layer of indirection: one can reach addresses that are not on the global forwarding table by mapping them to addresses that are on the table. This is the main idea behind the Map-and-Encap [13] proposal, which has been adopted in several specific designs, including 8+8 [14], LISP [15], ILNP [16], and APT [17], to name a few.

Building upon the Map-and-Encap idea, in this paper we propose a solution, dubbed SNAMP (Secure NAmespace MaPping), to address NDN's routing scalability concern. SNAMP enables the network to forward all interest packets towards the closest data even when not all data name prefixes are present in the global routing table. Data whose name prefixes do not appear in the global routing table can be retrieved using a securely mapped set of globally routed name prefixes, as exemplified in Fig. 1. The mapping information from a name to its globally routed prefixes will be maintained in and looked up from a distributed mapping system (see Section II-C for NDNS [18]). We believe that in many or even majority of the cases, data in NDN network can still be retrieved without resorting to SNAMP mechanisms: when the data is from local data producers, when the name prefixes *are* in the global routing table, or when the data are already in nearby caches. Therefore, SNAMP is designed to perform the name lookup step only when necessary.

Although SNAMP is still work-in-progress and yet to be implemented, we believe it is time to share its design for two reasons. First, this design has gone through several iterations over the last two years and we have learned a few important lessons through this process (see Section III); given there is a growing NDN research community, we believe these lessons are worth sharing. Second, we would like to solicit feedback and criticism about the SNAMP design from the community at large. This paper also serves as our call for collaborations from interested parties in further SNAMP development efforts.

978-1-4673-7131-5/15/$31.00 ©2015 IEEE

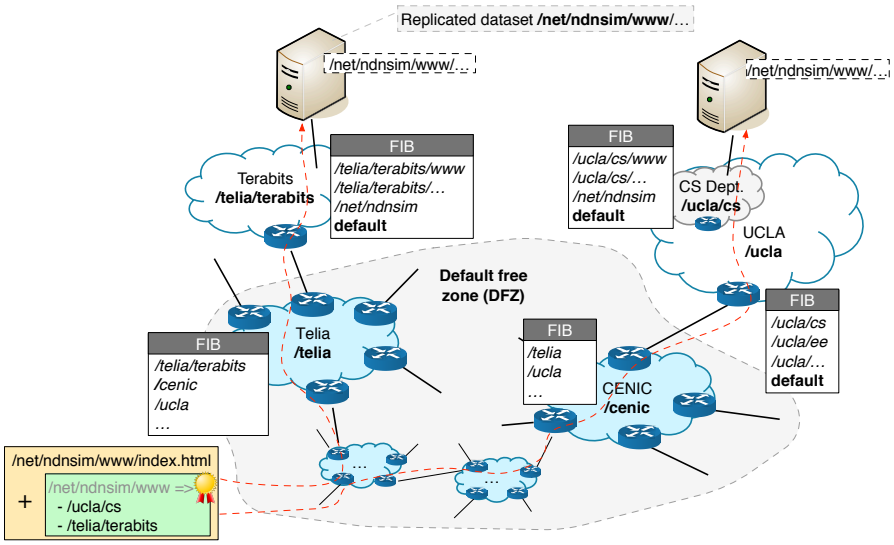

Fig. 1. Example of FIB state in an NDN network with Map-and-Encap: the "`/net/ndnsim/www`" dataset is replicated at Terabits and UCLA Computer Science Department, "`/net/ndnsim/www/index.html`" data can be reached directly inside Terabits and UCLA CS network, and can be reached globally if "mapped" to "`/telia/terbits`" or "`/ucla/cs`" prefix

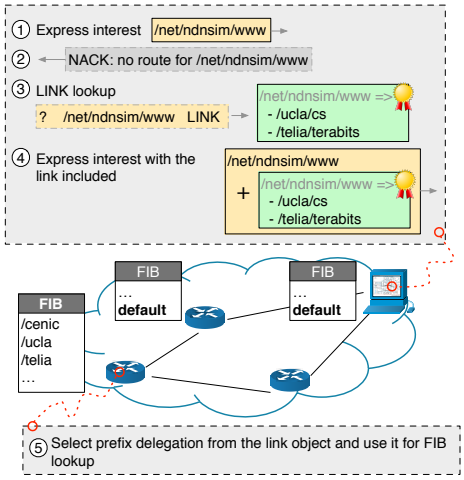

Fig. 2. Overview of map-and-encap NDN process

II. Design of the Secure Namespace Mapping

A. Overview

Different from today's TCP/IP architecture where application data names are used by applications only, while IP addresses are used for network packet delivery, data names in an NDN network are used directly in packet delivery. Data consumers send *interest* packets with the names of the content to be fetched, and data producers reply with *data* packets carrying the matching names. Routers perform routing computation, packet forwarding, and data caching all based on names. We assume routers run a routing protocol (e.g., NLSR [19] or some other name-based extension of OSPF or BGP) to build name-based forwarding tables, but do not assume any specific protocol in this paper.

To control the size of the global forwarding information base (FIB) at routers, one can apply a map-and-encap approach [13] and keep only a manageable subset of prefixes in the default-free zone (DFZ). In the rest of the paper, we will call these prefixes "globally routed prefixes". We intentionally leave the discussion about which prefixes should or should not be in the DFZ out of this paper, as these questions will be determined by the popularity of the data, network operational practices, and the tradeoffs between network functionality and the cost which will change over time as the technology continues to advance.

When a data name does not have its prefix in the DFZ, an interest packet carrying that name can still be directed toward the data, provided that the interest packet includes one or more globally routed prefixes of the network(s) through which the requested data can be retrieved. For example, Fig. 1 shows that, although "`/net/ndnsim/www/index.html`" cannot find a matching prefix in the DFZ FIB, interest packets carrying this name can be forwarded in the DFZ using FIB information for "`/ucla/cs`" or "`/telia/terabits`" prefixes. Once inside the local network environments (the UCLA CS department or the Terabits network), the data can be retrieved by its name directly.

Fig. 2 presents an overview of the proposed NDN map-and-encap process. In the following sections, we first briefly describe two important components in our design, the *link* object (Section II-B) and link discovery (Section II-C), and then describe the SNAMP-enabled NDN interest forwarding process in more detail.

B. The Link Object

In SNAMP, if a data producer wants to make data available globally but its prefix is not in the DFZ, the producer needs to establish an association between the name prefix (e.g., "`/net/ndnsim/www`") and the globally routed prefixes of its Internet service providers (e.g., "`/telia/terabits`" and

"/ucla/cs" in our example). We call this association a *link*. By creating and signing the link object, the owner of the original namespace N_o delegates its namespace to a set of namespaces $N_1, \ldots N_n$, essentially endorsing that the data under namespace N_o can be retrieved if interest is forwarded towards $N_1, \ldots,$ or N_n.

A link object is simply a piece of named data. The specific design of its naming and content is yet to be finalized. Preliminarily, we define the name of the link object to be N_o with a special mark (to avoid confusion with other types of data under the same name), and the data portion to be a list of delegated namespaces in some priority order.

C. Discovery of the Prefix Delegation Set

Together with the growing deployment of DNS Security Extensions (DNSSEC), DNS-Based Authentication of Named Entities (DANE) [20] provides an attractive means to use DNSSEC infrastructure to store and sign keys and certificates that are used by today's applications. Because the entities that vouch for the binding of public key data to DNS names are the same entities responsible for managing the DNS names in question, DANE restricts the scope of assertions that can be made by any entity, thus embodies the security "principle of least privilege."

Inspired by DANE, we have developed NDNS (DNS for NDN) [18], a scalable federated database system that retains these DNS/DANE properties and aims to serve a similar purpose in the NDN world. One of the usages is to serve prefix delegation maintenance and lookups.

Whenever needed, the owner of a namespace N_o can store its link objects in NDNS. Others can look up N_o's name to find its link using iterative or recursive resolution process [18] similar to the DNS query process. During the iterative resolution, either a dedicated caching resolver on behalf of the consumer or the consumer itself retrieves a set of link objects, gradually discovering the delegation namespace, one level at a time. In our example in Fig. 3, the process starts with the retrieval of the delegation information for "/net" NDNS namespace, followed by the retrieval of information about the "/net/ndnsim" NDNS namespace, and concluded by the retrieval of the "/net/ndnsim/www" namespace delegation (see [18] for more details).

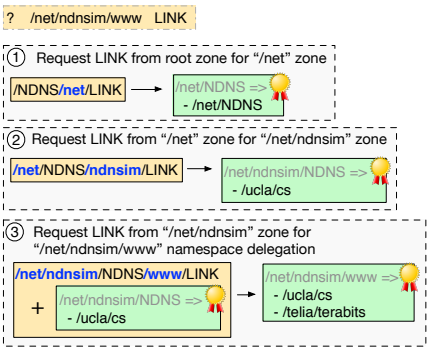

Fig. 3. Example of iterative NDNS query (performed by the recursive NDNS resolver on behalf of the consumer or by the consumer himself)

Note that NDNS requires only "/NDNS" prefix to be present in the DFZ FIB, pointing towards multiple replicas of the root NDNS server. This guarantees that all requests to the NDNS root zone can be answered. However, if all of the top level domain names (e.g., "/com/NDNS" and "/net/NDNS", etc.) are present in the DFZ, then the resolution process can start from the top level domains.

D. Retrieving Data with SNAMP

SNAMP is transparent from the application's perspective: the consumer applications send out ordinary NDN interests for the data, and the producer applications publish data in the namespace they own. When an expressed interest happens to carry a globally routed prefix (e.g., "/ucla/cs/www/index.html"), it can fetch the data either from a local cache or will be forwarded across the Internet towards the data producer. If the interest carries a name that is not present in DFZ (e.g., "/net/ndnsim/www/index.html"), it still can bring back data if the data happens to be in a local cache along the way, otherwise the interest reaches the first router that does not have "default" route and cannot be further forwarded. This is the place where SNAMP kicks in (steps 2–5 in Fig. 2).

A default-free router at the network edge will respond to this unroutable interest with a network NACK [21], indicating that it does not have a route for the interest's name and needs more information to forward the interest. This NACK eventually propagates back to the consumer node, and the local NDN forwarder for the consumer application will retrieve the link object and verify the validity of the delegation by checking the signature of the link object.

After the link object is retrieved and verified, the node will embed it into the original interest (see Section II-E) and send it out again. When this modified interest reaches the first router that cannot find a matching FIB entry for the interest name, the router will extract the prefix delegations from the attached link object (see Section II-E), select the best candidate (e.g., by using routing cost or based on previous traffic measurements), and forward the interest based on this selection.

Note that in the above procedure, routers that do not have a FIB entry for the interest name may need to perform multiple additional FIB lookups to determine the best namespace delegate for further forwarding (e.g., based on the routing cost). Optionally, the first default-free router may record its decision in the forwarded interests, e.g., by putting the index for the selected delegate in the optional field inside the interest. Downstream routers can then rely on this pre-recorded selection, unless the router is willing to do the selection again or when a problem is detected. Also note that every router can elect to verify the validity of the attached link object, so that interests are forwarded only when they carry a valid link object (i.e., the name of the link object must match the prefix of the interest name and has a valid signature), and are only forwarded toward the legit delegated prefixes. Routers can store verified link objects locally so that they do not have to verify the same link objects carried in subsequent interests.

Eventually, the interest either reaches a cache that can satisfy it or propagates down to one of the data producers. The cached or produced data is then returned to the original requester using standard NDN data forwarding mechanisms: following the state created by the forwarded interests.

E. Attaching Link Objects to Interests

In order to attach a link object to an interest packet, we propose to extend NDN-TLV interest packet specification [22] by including two additional optional fields (Fig. 4): "`Link`" to carry the link object and "`SelectedDelegation`" to carry the delegation namespace (its index inside the link object) chosen by the previous hop.

Note that after the prefix delegation is embedded into the interest, this interest can be effectively forwarded toward the data producer across the global Internet. It does not, however, mean that such an interest will always reach the producer; it may well hit a cache along the way and bring the data back from the cache.

III. Discussion

The SNAMP design started back in 2012 [23]. Since then the design has gone through several iterations and we have learned a few lessons. The current design, as described in this paper, includes a small change in the interest packet format and interest processing logic at NDN routers, which we believe should not expose any new critical vulnerabilities, while providing means to keep the global routing table under control and preserving all benefits of the NDN architecture. Note that the design is by no means final and may be modified further. In this section we share some of our design lessons and tradeoff considerations.

A. All Critical Information Must Be Signed

[23] describes our first SNAMP design. It differs from the current design mainly in three aspects. First, it did not spell out exactly where and when the mapping lookup step should be taken. Second, the *forwarding alias* specified only one namespace delegation. Third, and most importantly, the *forwarding alias*, which provides equivalent information as the *link*, is not secured—this unsecured mapping raised a serious concern as any router along the way could hijack an interest by changing its forwarding alias. This concern delayed publication of [23] by a year. Now we developed the link object to secure the name delegation.

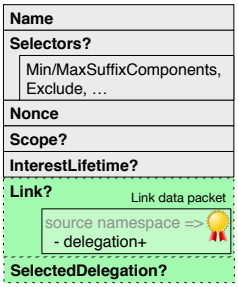

Fig. 4. New "Link" field in interest packet

B. Selection of Prefix Delegation

In many cases, especially for multi-hosted and/or multi-homed data producers, the link object will contain a set of delegations for the data name. We specifically designed SNAMP to defer the selection decision to the place where the decision can be made by informed parties—the routers that do not have a FIB entry for the interest name will use the link object and information from the routing protocols and data plane performance information maintained by NDN's adaptive forwarding plane [21] to forward the interest towards the closest data replica.

C. Modifiable Component in Interests

In Section II-E we introduced two optional fields to the interest packet. One of them ("`SelectedDelegation`") can be modified by the first router as a means to reduce the look up overhead for subsequent routers along the path. There is an ongoing discussion regarding whether NDN packets should, or should not, carry any modifiable components; and if such fields are needed for improving packet delivery performance, where to put them if not in the NDN packet format. One possibility is to put them in an adaptation layer below NDN.

D. Cache Poisoning

One commonly asked question about SNAMP is whether it introduces new vulnerabilities regarding cache poisoning. In other words, does it allow an attacker to direct an Interest to a place chosen by the attacker so that he/she can inject an invalid data packet into the caches along the path traveled by the interest? The proposed design modifies the way interests are forwarded by routers that do not have a corresponding FIB entry for the interest's name. Instead of dropping this interest, the router consults the attached link object to select a delegation to further forward the interest. Since the attached link object carries the cryptographic signature of the interest namespace owner, the link object cannot be replaced without detection. Therefore, the attacker cannot easily direct an interest towards an arbitrary place as long as routers check the validity of the link objects.

To reduce the risk of cache poisoning without requiring routers to verify link objects carried in interests, additional restrictions can be imposed on cache look-up. When a data packet is retrieved by an interest that carries a link object, the cache can store both the link object and data packet together. The cache can then impose the restriction that only interests carrying the *same* link object can be satisfied with the specific instance of the data item. This requirement increases cache processing cost, but ensures that even when routers do not verify link objects, the system does not allow attackers to subvert legitimate data retrieval because there is only one valid link object for the given data name.

Note that even when an invalid data packet is injected into the system, it does not mean that this packet will be cached, will not be quickly evicted, or that the consumers will fail to retrieve a valid data. Caches and consumers can check signature of data packets. In addition, the "exclude" capability

built into the NDN architecture provides a mechanism for consumers to avoid undesirable data. Furthermore, the Internet has a rich topological diversity, so the correct version of the data can be fetched over alternative path(s).

E. Alternative Design: Interest Encapsulation

An alternative design to incorporate globally routed names into interests to travel through DFZ is to prepend specific prefix delegations from the link objects to interest names. For example, to retrieve data "`/net/ndnsim/www`" across DFZ, it can be requested by an interest for "`/ucla/cs + /net/ndnsim/www`" or "`/telia/terabits + /net/ndnsim/www`" names. In fact this was our primary design choice for a long time, since it requires no changes in the processing logic of NDN routers. However, extensive discussions discovered a number of critical issues that led us to move away from this approach.

The first issue we ran into is that after an interest with the prepended link reaches the data producer, the producer faces two conflicting goals in returning the original data packet to the requester:

- the returned data packet name must match the interest with the encapsulated prefix delegation (e.g., "`/ucla/cs + /net/ndnsim/www`");
- the original data object with name "`/net/ndnsim/www`" must stay intact as the cryptographic signature binds the name and content.

One way to resolve this conflict is to encapsulate the original data packet in a new data packet with concatenated name: (1) the inner packet contains the original data name "`/net/ndnsim/www`", content and the signature that binds the two; (2) the outer packet contains the concatenated name, the inner packet as its content payload, and its own signature. However, this leads to the same data having multiple names and requires producer applications to take care of multiple prefixes under which data can be retrieved. This requires applications to discover and maintain current network attachments, a contradiction to the principles of data-centric communication.

The second issue concerns who should make the decision about which prefix delegation to prepend to an interest's name, since only one of the delegations can be used in such approach. If consumer applications have to take care of multiple prefixes under which data can be retrieved, they will likely lack an objective measure to select one delegation versus another. On the other hand, routers (especially those inside DFZ) have such information, e.g., in a form of the routing cost provided by the routing protocols. However, they will not have the ability to choose, given the prefix prepending is done by the consumer.

One way to leave the decision to the network is to require edge routers to automatically prepend the namespace delegations to interests before they enter DFZ and strip out the delegations when interests reach the producer networks (e.g., UCLA CS network in Fig. 1). However, this direction significantly increases system complexity and opens another set of performance and security problems. Routers at the edges will have to rewrite interests, while maintaining the correspondence between the original and the rewritten one. There is also a need for a special configuration on producer-side edge routers to detect that for some interests a prefix needs be stripped and for others should not; in addition, these routers also need to sign the encapsulated data packets. Finally, there is a question about the impact on NDN name discovery mechanism, i.e., whether selectors carried by interest packets will still work. For example, NDN allows excluding data packets based on implicit digest [22], which is computed over the wire format of the original data packet. If a data packet the consumer sees is different from the packet that actually traverses inside the DFZ, the consumer will lose the ability of excluding that packet, e.g., when it detects a compromised signature. In short, naming is the most critical component in NDN, any name change can lead to a host of complex issues.

IV. Related work

Routing scalability has long been a recognized problem of the present Internet [11]. A number of currently enforced regulations, e.g., limiting the maximum prefix size in the global routing table, keep the problem under control but do not eliminate it. The proposed solutions to completely eliminate the scalability problem can be categorized into two groups: namespace elimination and namespace separation [24]. The first group contains proposals that call for an extended use of multiple provider-dependent addresses, while upper layers (transport layers [25], [26] or a shim layer between IP and TCP [27]) need to take care of managing multiple addresses within a single connection. Proposals in the second group call for clear separation between addresses that appear in the global routing table and those addresses (or names) that are used by end-hosts ([13], [24], [28], [29], [30]), while some additional service (e.g., DNS) is used to map end-host addresses to (a set of) routable addresses.

Our proposal is in the same spirit as the map-and-encap approach [13] proposed for IP, but our design is consistent with NDN's name-based data retrieval model: (1) interests are sent by data consumers towards data producers, leaving a trail for returning data packets; and (2) an interest can retrieve data as long as the names match, even if the interest does not reach the producer.

Identifier-Locator Network Protocol (ILNP) [31] is a solution closely related to the proposal in this paper. In the current Internet, IP addresses are overloaded with the functionality of network locators and host identifiers, leading to many existing problems with application session maintenance as the network topology changes. ILNP explicitly "untangles" usages of IP addresses at different layers by mandating the use of separate network locators for packet forwarding, host identifiers for transport sessions, and DNS names within applications (e.g., not possible to use IP address in a WEB browser, instead of the domain name). Similar to our proposal, ILNP relies on the existing DNS/DNSSEC deployment to map from application-level domain names to node identifiers, which are then mapped to the network locators.

There are several notable differences between our proposal and the identity-locator separation approach adopted by ILNP for IP. Although in our proposal we have a similar separation between names that can be directly routed and names that require mapping, both types of names are from the same namespace and there are no clear boundaries between the two. Within different context, the same name can belong to different categories, e.g., our example "`/net/ndnsim`" website can be directly reachable within the hosted network, while needs mapping in the global context.

An alternative way to solve the routing scalability problem is to replace the conventional routing system with, as an example, geometric routing ([32], [33], [34]); we are actively exploring the hyperbolic routing approach for NDN. While this method does not require a global routing table, it still requires an additional mapping service to map names to hyperbolic coordinates. In this regard, the solutions are conceptually similar, but there is still a question about how well hyperbolic routing can work and how it can handle existing complex routing policies between ISPs.

V. Conclusion

In this paper, we proposed the application of the map-and-encap idea to scale NDN routing, where some name prefixes are present in the global routing table and the rest can be mapped to the globally routable prefixes using NDNS as a mapping service. The proposed secure namespace mapping (SNAMP) mechanism fully preserves NDN architecture features, is transparent to the consumer and producer applications, and is automatically enabled only when needed. In addition, we also shared the lessons we learned through the design exercises, in particular the necessity of securing all critical information and the caution against modifying data names.

Acknowledgment

This work was supported in part by the NSF grants CNS-1040868, CNS-1040036, CNS-1039615, CNS-1345318, CNS-1344495, and CNS-1345142. Junxiao Shi introduced the idea of cache processing restrictions to mitigate cache poisoning attacks described in Section III-D.

References

[1] D. Cheriton and M. Gritter, "TRIAD: A new next-generation Internet architecture," 2000.

[2] T. Koponen et al., "A data-oriented (and beyond) network architecture," in *Proc. of SIGCOMM*, 2007.

[3] S. Tarkoma, M. Ain, and K. Visala, "The publish/subscribe Internet routing paradigm (PSIRP): Designing the future internet architecture," *Towards the Future Internet*, 2009.

[4] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of information (NetInf) – an information-centric networking architecture," *Computer Communications*, vol. 36, no. 7, pp. 721 – 735, 2013.

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

[6] L. Zhang et al., "Named data networking (NDN) project," NDN, Tech. Rep. NDN-0001, October 2010.

[7] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM Computer Communication Reviews*, July 2014.

[8] A. Baid, T. Vu, and D. Raychaudhuri, "Comparing alternative approaches for networking of named objects in the future Internet," in *Proc. of NOMEN*, 2012.

[9] A. Narayanan and D. Oran, "NDN and IP routing: Can it scale?" Proposed Information-Centric Networking Research Group (ICNRG), Side meeting at IETF-82, November 2011.

[10] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB workshop on routing and addressing," RFC 4984, 2007.

[11] A. Afanasyev, N. Tilley, B. Longstaff, and L. Zhang, "BGP routing table: Trends and challenges," in *Proc. of High Tech. and Intell. Systems conf.*, 2010.

[12] Arbor networks, "Worldwide infrastructure security report," http://www.arbornetworks.com/research/infrastructure-security-report, Volume VII, 2011.

[13] S. Deering, "The map & encap scheme for scalable IPv4 routing with portable site prefixes," *Presentation Xerox PARC*, 1996.

[14] M. O'Dell, "8+8—an alternate addressing architecture for IPv6," Internet draft (draft-odell-8+8-00), 1996.

[15] D. Farinacci, "Locator/ID separation protocol (LISP)," Internet draft (draft-farinacci-lisp-00), 2007.

[16] R. Atkinson, S. Bhatti, and S. Hailes, "ILNP: mobility, multi-homing, localised addressing and security through naming," *Telecommunication Systems*, vol. 42, no. 3, 2009.

[17] D. Jen, M. Meisel, D. Massey, L. Wang, B. Zhang, and L. Zhang, "APT: A practical tunneling architecture for routing scalability," UCLA Comp. Sc. Dep., Tech. Rep. 080004, 2008.

[18] A. Afanasyev, "Addressing operational challenges in Named Data Networking through NDNS distributed database," Ph.D. dissertation, UCLA, September 2013.

[19] A. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: named-data link state routing protocol," in *Proc. of SIGCOMM Workshop on Information-Centric Networking*, 2013, pp. 15–20.

[20] P. Hoffman and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA," RFC 6698, 2012.

[21] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in Named Data Networking," *ACM Computer Communication Reviews*, vol. 42, no. 3, pp. 62–67, July 2012.

[22] NDN Project, "NDN Packet Format Specification," Online: http://named-data.net/doc/ndn-tlv/, 2014.

[23] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and L. Zhang, "Scaling ndn routing: Old tale, new design," NDN, Technical Report NDN-0004, July 2013. [Online]. Available: http://named-data.net/techreports.html

[24] D. Jen, M. Meisel, H. Yan, D. Massey, L. Wang, B. Zhang, and L. Zhang, "Towards a new internet routing architecture: Arguments for separating edges from transit core," in *Proc. of HotNets*, 2008.

[25] P. F. Tsuchiya, "Efficient and robust policy routing using multiple hierarchical addresses," in *Proc. of SIGCOMM*, 1991.

[26] M. Handley, D. Wischik, and M. B. Braun, "Multipath transport, resource pooling, and implications for routing," Presentation at IETF-71, July 2008.

[27] E. Nordmark and M. Bagnulo, "Shim6: Level 3 multihoming shim protocol for IPv6," Internet draft (draft-ietf-shim6-proto-09), 2007.

[28] D. Massey, L. Wang, B. Zhang, and L. Zhang, "A scalable routing system design for future internet," in *Proc. of SIGCOMM IPv6 and the Future of the Internet workshop*, 2007.

[29] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, "The design and implementation of an intentional naming system," in *SIGOPS Operating Systems Review*, vol. 33, 1999.

[30] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in content-oriented architectures," in *Proceedings of SIGCOMM Workshop on ICN*, 2011.

[31] R. J. Atkinson and S. N. Bhatti, "Identifier-locator network protocol (ILNP) engineering considerations," RFC 6741, November 2012.

[32] F. Papadopoulos, D. Krioukov, M. Boguna, and A. Vahdat, "Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces," in *Proceedings of IEEE INFOCOM*, 2010.

[33] M. Boguna, F. Papadopoulos, and D. Krioukov, "Sustaining the Internet with hyperbolic mapping," *Nature Communications*, vol. 1, no. 62, 2010.

[34] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguna, "Hyperbolic geometry of complex networks," *Physical Review E*, vol. 82, 2010.