NDN, Technical Report NDN-0004, 2015. http://named-data.net/techreports.html
Revision 1: July 18, 2013. http://named-data.net/techreport/ndn-0004-1-scaling-ndn-routing.pdf
Revision 2: January 8, 2015. http://named-data.net/techreport/ndn-0004-2-scaling-ndn-routing.pdf
Revision 3: February 3, 2015

Map-and-Encap for Scaling NDN Routing

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. One key component of the architecture is direct use of application names to route requests for data, which raises a concern about routing scalability in an NDN network. In this paper we apply a well-known concept of *Map-and-Encap* to NDN routing to keep the size of the global routing table under control. More specifically, we securely map application names to one or more globally routed names whenever needed to inform the forwarding system of the whereabouts of the requested Data. This solution enables NDN routing to scale with Internet's well understood routing protocols and operational practice, while keeping all the benefits of the new architecture.**

I. Introduction

As Internet applications become increasingly data-centric, a number of new network architecture designs [1], [2], [3], including Named Data Networking (NDN) [4], [5], [6], have proposed a data-centric communication paradigm, which in part requires the network routing system to deliver packets directly on data names. In an NDN network, interest packets carry hierarchical names of desired data (e.g., "`/net/ndnsim/www`") that are used by routers to forward the interests towards the requested data. By naming data directly and binding the name and data by a cryptographic signature, NDN provides a number of benefits including data security, in-network caching, built-in multicast support, 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 routing ([7], [8]). Given that the number of data names is unbounded, can one keep the size of name-based NDN routing table under control?

Routing scalability concern is not new. Although IP address space is finite, IP routing scalability has been considered one of the major challenges since the early days of Internet deployment. One basic approach to keeping global IP routing table under control is address aggregation: ideally 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 [9], network-layer traffic engineering and load balancing [10], and mitigation of DDoS attacks and prefix hijacks [11], 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: addresses whose reachability do not follow topological hierarchy can be reached using addresses that follow the hierarchy. This is the main idea behind the Map-and-Encap [12] proposal, which has been adopted in several specific designs, including 8+8 [13], LISP [14], ILNP [15], and APT [16], to name a few.

Building upon the Map-and-Encap idea, in this paper we propose a solution, dubbed MAEN (Map-and-Encap for NDN), to address NDN's routing scalability concern. MAEN enables the network to forward all interest packets toward the closest data even when not all the 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 set of globally routed name prefixes (Figure 1), as described in Section II-D. The mapping information from a name to its globally routed prefixes can be maintained in, and looked up from, a distributed mapping system (NDNS [17]) as described in Section II-C. Since data can be retrieved directly using the original names in many or most cases, for example 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, MAEN performs the name lookup step only when necessary and only by end consumers.

Although MAEN 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 improvements over the last two years and we have learned a few important lessons through the 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 MAEN design from the community at large. This paper also serves as our call for collaborations from interested parties in further MAEN development efforts.

II. NDN Map-and-Encap Design

A. Overview

Applications name their content. Different from today's TCP/IP architecture where application data names are first resolved to IP addresses before communication can start, in an NDN network data names 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 [18] or some other name-based extension of OSPF or BGP)

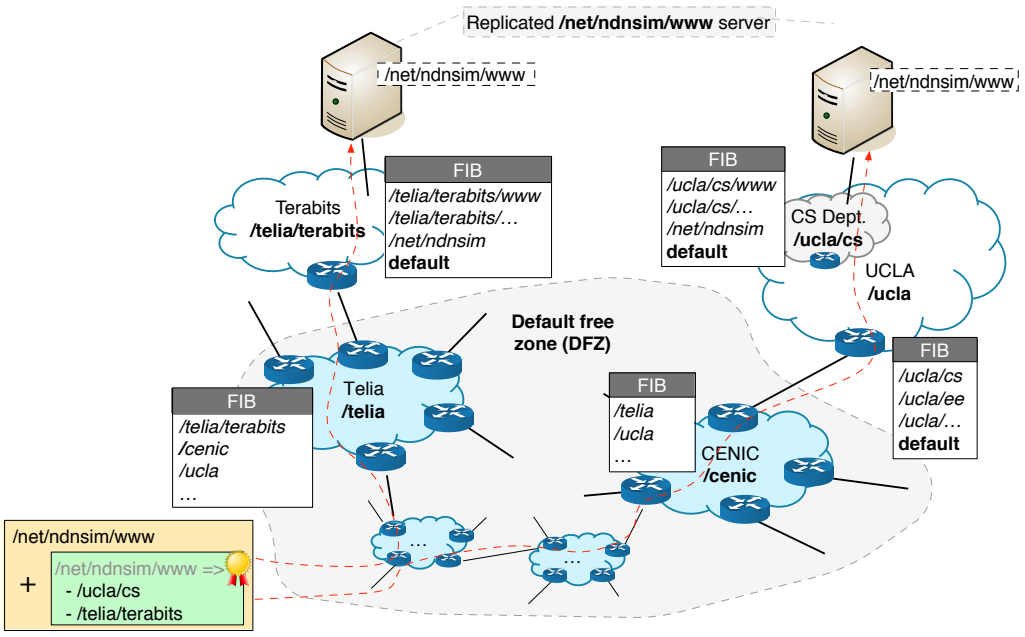

Fig. 1. Example of FIB state in an NDN network with Map-and-Encap: the /net/ndnsim/ server is replicated at Terabits and UCLA CS department, "/net/ndnsim/www" data can be reached directly inside Terabits and UCLA CS networks, and can be reached globally if "encapsulated" under "/telia/terbits" or "/ucla/cs" prefix

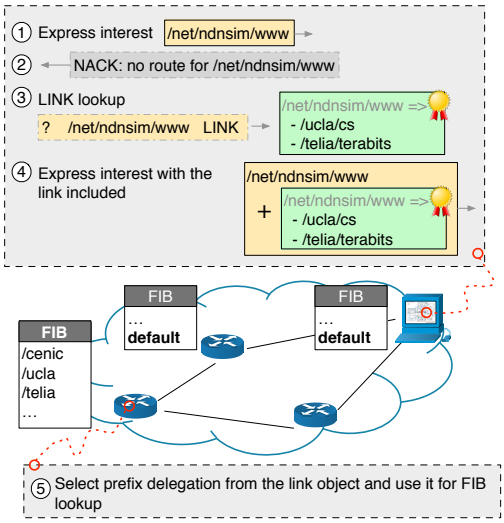

Fig. 2. Overview of map-and-encap NDN process

to build name-based forwarding tables, but do not assume any specific protocol in this paper.

Assuming routers cannot hold the name prefixes for all the data, one can control the global routing table size by applying a map-and-encap approach [12] to keep only a manageable subset of prefixes in the default-free zone (DFZ) routing tables. In the rest of the paper, we will call these prefixes "globally routed prefixes". We intentionally leave discussions of which prefixes would be, or would not be, in the DFZ out of this paper, as these questions will be determined by the popularity of the named data, the tradeoff between cost and network functionality, and network operational practices.

When a name does not have its prefix in the DFZ, an interest packet carrying that name can still be directed toward the data producer site(s) if the interest packet can include one or more globally routed prefixes of the network(s) that connect to the data producer. For example, Figure 1 shows that, although "`/net/ndnsim/www`" cannot find a matching prefix in the DFZ FIB, interest packets carrying this name can be forwarded in the DFZ using "`/ucla/cs`" or "`/telia/terabits`". Once inside the local network environments (the UCLA CS department or the Terabits network), the data producer can be directly reached.

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

B. The Link Object

In a map-and-encap system, if a data producer's prefix is not in the DFZ forwarding table, it needs to establish an association between its own name prefix (e.g., "`/net/ndnsim`") 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*.

A link represents a namespace delegation. That is, the owner of namespace-1 (e.g. "`/net/ndnsim`") delegates its namespace to namespace-2 (e.g. "`/telia/terabits`" and

"/ucla/cs"), thus interests carrying names under namespace-1 can be forwarded to namespace-2. This delegation must be authorized by the owner of namespace-1. In our current design, a link object is simply a piece of named data that can be fetched as any other data. The name of this object is namespace-1 with a special mark indicating that it names a link object (to avoid confusion with other types of data under the same name), the data portion is a list of namespaces namespace-1 is delegated to, and this object is signed by the key of namespace-1 to secure the delegation.

C. Discovery of the Prefix Delegation Set

Together with the growing deployment of DNS Security Extensions (DNSSEC), DNS-Based Authentication of Named Entities (DANE) provides an attractive means to use DNSSEC infrastructure to store and sign keys and certificates that are used by 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) [17], 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, stores its link objects in NDNS. Others can look up N_o's name to find its link using iterative or recursive resolution process [17] similar to DNS query processes. 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 specializing delegation namespace. For 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.

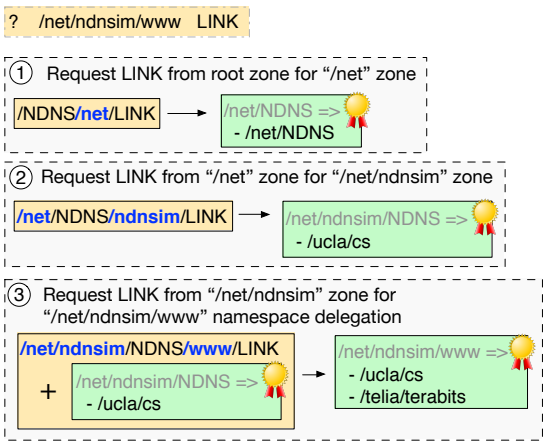

Fig. 3. Example of iterative NDNS query (performed by the recursive NDNS resolver on behalf of the consumer or by the consumer himself)

Note that NDNS requires only the "/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 root zone can always be answered. When necessary, the link object discovered in the previous iteration is used to reach NDNS servers at next level, so that one is always able to retrieve the requested data (see [17] for more details). If all of the top level domain names (e.g., "/com/NDNS" and "/net/NDNS", etc.) are also present in the DFZ, the resolution process can start from the top level domains.

D. Retrieving Data with Map-and-Encap

The map-and-encap process is transparent from the application perspectives: 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 carries a globally routed prefix (e.g., "/telia/www"), it fetches 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"), 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 the map-and-encap NDN extension kicks in (steps 2–5 in Figure 2).

A default-free router at the network edge will respond to this unroutable interest with a network NACK [19], indicating that it does not have a route for the interest's prefix and needs more information to forward the interest. This NACK eventually propagates back to the consumer node, and the local NDN forwarder (resides in the node) will retrieve the "link" object (see Section II-B) and verify the validity of the delegation by checking both the signature of the link object as well as the provider agreement links.

After the link object is retrieved and verified, the node will embed it into the original interest and express it. When this modified interest reaches the first default-free router, the router will extract 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. Schematically, the process is illustrated in Pseudocode 1.

Note that in the above procedure, all routers that do not have a FIB entry for the interest name would perform multiple FIB lookups to determine the best delegated candidate for further forwarding. Optionally the first default-free router may record its decision in the forwarded interests, i.e., putting the selected delegate in an additional field inside the interest. Downstream DFZ routers can then rely on this field, unless the router is willing to do the selection again or a problem is detected. Also note that every router can elect to verify the validity of the attached link object, so that forwarding based on the delegated prefixes are restricted only to those interests carrying a valid link object, i.e. the link object with a name that is a prefix of the interest name and having a valid signature.

Pseudocode 1 Interest Processing

```
1:  function PROCESS(Interest)
2:      Name ← Interest.Name
3:      LinkObject ← Interest.LinkObject
4:      if Data ← Cache.Find(Name) then
5:          Return(Data)
6:      else if PitEntry ← PIT.Find(Name) then
7:          Record(PitEntry, Interest)
8:      else if FibEntry ← FIB.Find(Name) then
9:          Forward(FibEntry, Interest)
10:     else
11:         FibEntry ← ∅
12:         for each Name_d in LinkObject do
13:             FibEntry ← Best(FibEntry, FIB.Find(Name_d))
14:         end for
15:         if FibEntry then
16:             Forward(FibEntry, Interest)
17:         else
18:             Drop Interest (Interest cannot be satisfied)
19:         end if
20:     end if
21: end function
```

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 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 [20] by including an additional optional "Link" field (Figure 4).

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.

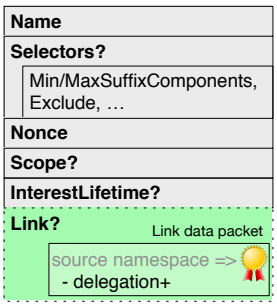

Fig. 4. New "Link" field in interest packet

III. Discussion

The MAEN design started back in 2012 [21]. Since then the design has gone through a few 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 a means to keep the global routing table under control and preserving all benefits of NDN architecture. In this section we share some of our design lessons and tradeoff considerations.

A. All Critical Information Must Be Signed

[21] describes our first MAEN design. It differs from the current design mainly in two aspects. First, it did not spell out exactly who and when the mapping lookup step should be taken as we were still unclear about the best choice. Second, and most importantly, the *forwarding alias*, which provides equivalent information as *link* does, 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. The concern delayed the publication of [21] by a year. Now we developed the link object to secure the name delegation.

B. Alternatives of Interest Encapsulation Formats

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 forward the interest "`/net/ndnsim/www`" through DMZ, it can be sent by carrying the "`/ucla/cs + /net/ndnsim/www`" or "`/telia/terabits + /net/ndnsim/www`". In fact this was our primary design choice for a long time, since it requires no change 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 prepended link, I_{prep} 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 lead to the same data having multiple names. In addition, producer applications will need to take care of multiple prefixes under which data can be retrieved. This requires applications to discover and maintain current network

attachments, which seem to contradict the principles of data-centric communication.

A second question is who should make a decision about which prefix delegation to prepend to an interest's name, since only one of the delegations can be used. If consumer applications have to take care of multiple prefixes under which data can be retrieved, they likely lack an objective measure to select one delegation versus another. Requiring applications to discover and maintain information about the current network attachments contradicts NDN's principles of data-centric communication. On the other hand, routers inside the network (especially those inside DFZ) have such information (e.g., routing cost or policy provided by the routing protocols), but it will not have the ability to choose, if the decision has been already made by consumer applications.

If we leave the decision to the network, the network must perform NAT equivalent function by automatically prepending links to interests before they enter DFZ and stripping them out before exiting, a direction that can significantly increase system complexity and raises another set of 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 on some interests a prefix needs be stripped and on others should not. Finally, there is a question of how NDN name discovery mechanism (selectors carried by interest packets) may work. For example, NDN allows excluding data packets based on implicit digest [20], 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 ability of excluding that packet, e.g., when it detects a compromised signature. Naming is the most critical component in NDN, any name change can lead to a host of complex issues.

C. 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 the system to defer the selection decision to the place where the decision can be made consciously, i.e., the first router that does not have a FIB entry for the interest name, using information from the routing protocols and data plane performance information maintained by NDN's adaptive forwarding plane [19].

D. Cache Poisoning

One commonly asked question about MAEN is whether it introduces new vulnerability regarding the cache poisoning. In other words, does it make easier to inject an invalid data packet for an arbitrary name into caches? 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 cache.

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" filter in interests provides a mechanism for consumers to avoid undesirable data; and, in general, the Internet has a rich topological diversity, so the correct version of the data can be fetched over alternative path(s).

IV. Related work

Routing scalability has long been a recognized problem of the present Internet [10]. 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. A common approach for solving this problem is through address aggregation, which requires address allocation to reflect the network topology, either directly or indirectly. Existing solutions can be categorized into two groups: namespace elimination and namespace separation [22]. The first group contains proposals that call for an extended use of multiple provider-dependent addresses, while upper layers (transport layers [23], [24] or a shim layer between IP and TCP [25]) need to take care of managing multiple addresses within a single connection. When establishing connection, upper layers may either obtain multiple addresses through DNS by mapping an application-requested domain name to a set of resource records, or negotiate during the connection handshake (Shim6). 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 ([12], [22], [26], [27], [28]), while some additional service (e.g., DNS) is used to map end-host addresses to (a set of) routable addresses. In both cases, addresses that appear in the global routing table are only addresses of ISPs, which are limited in number and easy to aggregate.

Our proposal is in the same spirit as the map-and-encap approach [12] 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) [29] is a solution closely related to the proposal in this paper. In the current Internet, IP addresses (both in IPv4 and IPv6) 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 global context.

An alternative way to solve the routing scalability problem is to replace the conventional routing system with, as an example, geometric routing ([30], [31], [32]); 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 a 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 MAEN mechanism fully preserves unique 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.

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] 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.

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

[6] 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.

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

[8] 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.

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

[10] 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.

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

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

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

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

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

[16] 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.

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

[18] 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.

[19] 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.

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

[21] 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

[22] 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.

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

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

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

[26] 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.

[27] 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.

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

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

[30] 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.

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

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