Globally Accessible Names in Named Data Networking

Niels L.M. van Adrichem and Fernando A. Kuipers
Network Architectures and Services, Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
{N.L.M.vanAdrichem,F.A.Kuipers}@tudelft.nl

***Abstract*—Information Centric Networking (ICN) paradigms aim at optimizing computer networks for information distribution. Named Data Networking (NDN) and its implementation CCNx propose a promising globally implementable ICN. Routing on names, however, may result in extremely large global routing tables. In this paper, we propose to confine the global routing table size by decoupling context-related names, such as domain names, from names routable within the network. By aggregating routable names to their topological location, the size of global routing tables decreases to the number of Autonomous Systems. Furthermore, mapping context-related names back to location-aggregated names using a directory service eases the process of sharing information on the ICN. The robustness of the network is further increased by employing dynamic multihoming without changing application names.**

I. Introduction

Host-to-host networks – such as IP networks – are designed to efficiently forward packets from one host to another over large networks, such as the Internet. However, practice shows that host-to-host networks are inefficient at distributing content on a large scale, since data will often duplicately travel along the same links and nodes. Optimizing for information distribution in IP is difficult due to applications explicitly requesting the network to deliver packets, therefore only enabling optimizations for host-to-host packet delivery. Lately, much work [1][2][3][4] has been done in creating Information Centric Networks (ICNs) that optimize for information distribution instead, by making the networks more aware of their function as information distributor.

Named Data Networking (NDN) [5] implements an Information Centric Network by forwarding packets based on the names of the content they request or carry, instead of the addresses of their requester and generator. By basing the packet forwarding across the network on the content name or description, we allow the network to be optimized for distribution of content. The most straightforward optimization is obtained by delivering duplicate requests from caches placed within the shortest paths from client to server. Duplicate requests can be detected by comparing the names of requested content to earlier delivered data and providing the results from memory accordingly.

NDN proposes a promising globally implementable Information Centric Networking technique:

- It is independent of underlying – if any – transport protocols.
- It deploys a clear three-table prefix match and forwarding mechanism, as discussed in section II.
- It offers new opportunities to Quality of Service and exploitation of multipath routes.

Unfortunately, NDN also knows problems that need to be solved before a global implementation becomes feasible:

(1) As described in [2], we need to contain the global routing table size resulting from name-based forwarding. Currently, over 240 million domain names[1] [6], versus more than 42.000 ASes and 438.000 BGP propagated IP subnets [7], exist. Given that the size of the BGP routing tables is already problematic [8], an increase of a factor 550 or more introduced by name-based routing cannot be solved without further aggregation of names.

(2) Another problem is the difficulty for end-users to share information on the network. Currently, sharing information requires configuring the NDN enabled topology discovery mechanism OSPFN [9]. We need a mechanism enabling end-users to dynamically access and share information on the ICN without the need to configure topology discovery mechanisms.

Where previous research mainly focused on speeding up lookups from the routing tables using hash tables [10][11], we aim to reduce the size of the problem. We will first describe the basic techniques of NDN in section II, followed by our proposal of decoupling context-related names from globally routable names in section III. In section IV, we discuss the implications on naming and topology discovery in the different routing domains[2]. Section V describes our proposed mapping technique and shows that our implementation realizes such a mapping mechanism for NDN, while maintaining the NDN features. In section VI we present our implementation, which dynamically generates aggregated names within LANs. Finally, section VII discusses previous research in this area.

II. Named Data Networking

In this section, we will briefly discuss how Named Data Networking (NDN) [5] and its implementation CCNx form an Information Centric Network (ICN) by using a route-by-name principle. Opposed to source and destination addresses in IP packets, NDN knows Interest and ContentObject packets identified by names composed of one or more name components (e.g., Alice could share her photos using the name

[1] Excluding the existence of NDN routable subdomains and subnames.

[2] Inter-domain, intra-domain and local area networks.

/alice.eu/photos). Whenever a client requests information, it sends out an Interest containing a name describing the desired information. Subsequent nodes forward that Interest hop-by-hop along the shortest path to the generator responsible for that name until it either reaches a node that has a cached copy satisfying the description of the Interest, or it reaches the content generator. The resulting data is encapsulated in a ContentObject and is forwarded along the exact reverse path back to the requester.

Functionally, each subsequent node maintains three tables:

1) The ContentStore (CS), containing cached copies of previously delivered data.
2) The Pending Interest Table (PIT), in which forwarded Interests and their originating faces[3] are stored.
3) The Forwarding Information Base (FIB), containing forwarding rules based on names instead of addresses.

Each incoming Interest is first compared to the content of the CS to determine if it can be fulfilled from cache. If not, the packet is compared to the PIT to prevent duplicate requests traveling down the network. Finally, the FIB is consulted to determine to which faces the Interest needs to be forwarded. The Interest and its originating face will be added to the PIT accordingly. Once an Interest hits a valid copy or a new ContentObject is generated, the PIT entries are used to forward the resulting ContentObject back to its original requester.

While IP prefix matches on subsequent bits, NDN prefix matches on subsequent name components. Each subsequent component adds a more restrictive requirement to the content (e.g., the name /alice.eu/photo/holidayPhoto prefix matches an Interest for /alice.eu/photo, as the requester did not specify the exact photo).

III. Recursive Name Aggregation

In order to keep the complexity of global routing tables within limits, we will decouple routable names from user-registered context-related names as suggested in [2]. Context-related names are mapped to one or more routable names describing the locations of the hosts responsible for generating the corresponding content. In this section, we will propose a structure for the routable names in different routing domains.

(1) Inter-domain routing should be based on single Name-Components describing the AS to which the generating host belongs. This bounds the size of global routing tables to the number of ASes in the network[4]. For example, global routing may occur using names such as /isp-name.net for providers and /company-name for enterprise companies maintaining their own AS.

(2) By using subnames of the AS specific name for nodes within an AS network, the AS only needs to set up node-specific routes internally without affecting global routing tables. For example, a server within an AS network may be named /isp.net/content-server1, which becomes globally accessible since the prefix /isp.net points towards the AS and the AS will route incoming Interests to the specific node internally. In-network routing tables will be as complex as they would be in IP networks, as the process of subnaming conforms to the process of IP subnetting.

(3) Local network nodes may use names aggregating to the name of their modem or access router as provided by their ISP, which in turn can be used for further subnaming without affecting the AS its routing tables.

The advantage of a topologically layered naming scheme is the fact that subnets can be dynamically generated and globally used without affecting the routing tables of the higher-level networks. NDN advantages – such as caching, multipath routing and the Strategy layer – are preserved since forwarding is still done based on globally unique hierarchical names. In section IV, we will discuss the implications in terms of naming and routing discovery for the identified routing domains.

Although the global name complexity can be contained, users prefer to work with context-related names. In IP, applications use the Domain Name System (DNS) [12] to translate the authoritative part of a (generally context-related) URL to a location in the form of an IP address. We, however, propose to use a mapping system similar to the Location Identifier Separation Protocol (LISP) [13] to have the network map context-related names to location-based names. Applications base communication and content retrieval on the context-related names, while networking nodes (such as the host CCNx daemon or the default gateway) rename Interests and ContentObjects from the context-related names to the location-based names and vice versa.

By mapping context-related names to routable names in the network, application communication takes place independent of the location, the number and the availability of end-points; in NDN one does not care who delivers the information nor the path it has taken as long as the delivered information is guaranteed to be authentic.

IV. Naming and discovery

In this section, we will discuss the naming of entities and how to find them by means of topology discovery.

A. Inter-Domain Routing

We propose to use AS-bound context-unrelated first name components for inter-domain routing. The first NameComponent can be either descriptive for the AS (e.g., /isp.as), contain the Autonomous System Number (e.g., /AS1103) or a variable-length byte-encoded integer containing the AS Number (as prefix-matching on integers is a less expensive operation than on strings). If all inter-domain routing is based on these first name components, the size of global routing tables is upper bounded by the number of ASes present on the Internet.

In order for ASes to forward and serve NDN traffic, we need to adapt the inter-domain topology discovery mechanism BGP [14] to propagate NDN compatibility and name information. Since the standardization of Multiprotocol Extensions for BGP-4 [15][16], it is possible to propagate the availability of

[3] The NDN equivalent of an interface. Besides Ethernet links a face can also include connections with applications and several other types of connections.

[4] Which, according to [7], is a magnitude of 10 smaller than the current number of propagated IP prefixes.

other address families than IPv4 and IPv6 in BGP. Narayanan et al. [17] are working on realizing such an implementation by introducing a new address family supporting NDN names, although they presently do not consider crossing NDN incompatible ASes.

De Clercq et al., in RfC 4798 [18], discuss a standard in which IPv6 is tunneled dynamically across (IPv6 incapable) IPv4/MPLS networks. NDN needs a mechanism with properties from both Narayanan et al. and RfC 4798 to realize connectivity across NDN incompatible ASes by using dynamically set-up tunnels. Where NDN however does not know IP-mapped names, we need to find other ways to insert tunnel end-points into the propagated names[5].

B. Intra-Domain Routing

Nodes within an Autonomous System can, in the location-aggregated naming scheme, be named at will as long as their name prefix matches the previously discussed AS-bound name that is globally propagated. Names consisting of concatenating the AS-bound name and an internally unique router identification, such as /isp.net/routerX, qualify. If an AS knows different layers in which subnetworks are connected it may represent those in the names of the nodes.

Internal routing discovery needs to occur in order to propagate the availability of names internally. Popular intra-domain routing discovery mechanisms, such as OSPF and IS-IS, need to be adapted to support the new address family introduced by the hierarchical naming structure of NDN.

OSPFN [9] is an implementation of such an adaptation using OSPF to propagate available NDN names in a network. By using OSPF its Opaque LSA options, the developers of OSPFN have designed a conveniently accessible mechanism taking advantage of an already existing infrastructure. However, the OSPFN daemon reuses the shortest paths computed by OSPF on the local Routing Information Base without considering whether each node within the network is NDN compatible.

We propose a mechanism similar to OSPFN that uses Opaque LSAs to spread NDN compatibility and names. When considering shortest paths between NDN compatible nodes, parts of the path that are NDN incompatible should either be omitted or crossed using IP encapsulated tunnels.

C. Local Area Network

We propose to base naming and routing of Local Area Networks (LANs) on the subnames of the access router (generally a cable- or DSL-modem) by which they are connected to their ISP. The ISP generates the name of the entrypoint and solves routing in its network, while users can use subnames aggregating to that name to share information from within the LAN.

Similar to IP's Dynamic Host Configuration Protocol, we need mechanisms to dynamically configure end-user devices and enable them to dynamically retrieve and share information on the NDN. CCNx-DHCP [19] already fulfils part of that duty by introducing a server-client model that preconfigures many clients on a network. Unfortunately, CCNx-DHCP does not enable end-users to also share information on the network. Currently, a user must propagate his name using OSPFN [9] into the network before it properly forwards Interests meant for him, which is undesirable. Furthermore, the local area network topology discovery usually lies out of the propagation scope of the AS, which prevents global propagation of the name.

In order to have users fully participate in the NDN[6], we propose to deploy a configuration-free routing discovery mechanism to find the border routers within the LAN connecting it to the ISP network. The shortest paths to those entrypoints are used as gateway forwarding rules, while the reverse path is used to form names aggregating back to the names of the entrypoints. This enables (both consumer and enterprise) Local Area Networks to form dynamically without the effort to administrate address ranges and routing tables. Section VI discusses our proposal and its implementation for recursive name aggregation in LANs.

V. Mapping

An important aspect of our proposal is how mappings between context-related and location-aggregated names are stored and retrieved. Given the fact that the mapping or translation of the names needs to occur on-the-fly, a very low time complexity is necessary for a successful implementation.

We propose to use the Domain Name System (DNS) to store mappings between context-related and location-aggregated names. Although DNS's worst-case time complexity is not guaranteed, [20] shows an average response time of 43 ms for conventional DNS compared to 350 ms response time for a peer-to-peer distributed DNS. Throughout the years, DNS has been used to map domain names to – among others – IP addresses. We propose to either extend DNS by adding a new type of record to store NDN mappings, or reuse one of the generalized service (SRV) or text (TXT) descriptions.

Where IP clients use the address retrieved by DNS to base their networking connections on, we propose a slightly different solution based on LISP-TREE [21] and its equivalent LISP-DDT [22]. Applications base their communication on the context-related names, while networking instances on either the hosts or gateways near them translate the Interests to their location-aggregated names[7]. The renamed Interests are forwarded accordingly to a host in the network of the content generator who can translate the name back to its original value.

Figure 1 shows a topology in which a local network maps the desired context-related name to three globally reachable end-points that forward the Interests to the responsible nodes.

A. Location independence and detecting broken paths

By basing the communication protocol between applications on context-related names, one allows the network to find another path to the content serving the application[8] independent

[5] For example by using a directory service or by assigning AS specific IP addresses that each AS should run tunnel end-points on.

[6] Meaning that they can both *access* and *share* content dynamically.

[7] Whose values are fetched using DNS.

[8] By selecting another location-aggregated name by which the serving host can be connected.

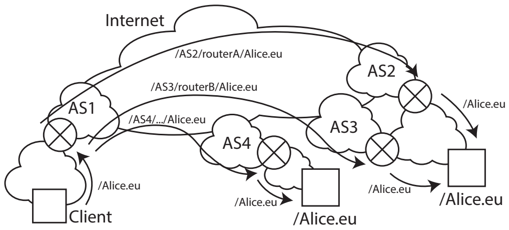
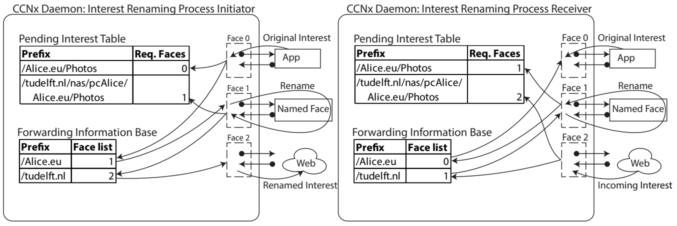


Fig. 1: Topology containing context-related routable networks connected over a globally routable network using mapping. Inter-domain forwarding occurs based on the first NameComponent, intra-domain on the second, while the LAN networks forward based on the context-related names.



(a) Interest Renaming Initiator (b) Interest Renaming Receiver

Fig. 2: The Interest renaming mechanism. After a ContentStore mismatch, context-related Interests are forwarded to the appropriate Named Face which re-expresses the Interests with the corresponding location-aggregated names.

of its location and without interrupting the application data flow.

While IP routing discovery selects the best of all paths to each destination and IP kernels merely forward packets along these paths, NDN multipath routes may coexist and Interests can be duplicated and sent along multiple paths. NDN introduces a strategy layer in which the performance of these different routes are evaluated. Based on this evaluation the forwarding engine can decide to send a larger portion of the Interests necessary to retrieve the complete file to the route behaving best (in terms of latency, throughput and packet loss), while maintaining the occasional Interest to monitor the state of the other routes.

Using our renaming mechanism implementation discussed in subsection V-B, we enable the strategy layer to balance between the available location-aggregated names – and thus paths – for any context-related name. Using the strategy layer, broken paths are detected and omitted without changing the mapping[9] or recalculating the network topology.

B. CCNx Renaming Implementation

In this section, we will discuss the details of implementing the described renaming from context-related names to location-aggregated names in CCNx. As shown in figure 2a, we add a new type of face – the named face – which renames Interests containing context-related names by appending the location-aggregated name of one of the receiving end-points to the front of the requested Name. At the receiving node depicted in figure 2b, a similar face removes the location-specific part of the name and forwards the Interest to the appropriate application or next-hop. As all intermediate names are stored in the Pending Interest Table with their respective sources, we assure that the returned ContentObject travels the renaming steps in reverse order.

Since each mapping from a context-related name to a location-aggregated name is expressed in the form of a distinct unique face and a route, multiple faces exist when multiple mappings are available resulting in a forwarding rule for each mapping. The CCNx strategy layer chooses the best of these routes – and thus mappings – over time and reverts to the other route when the path of the primary route breaks. This failover procedure is shown by experiment in [5]. Our freely available implementation [23] offers a clean and simple mechanism to insert mappings from context-free to location-aggregated names and back into an NDN. Once an Interest arrives for an unknown context-related name, the implementation queries the DNS system, which stores the mappings using TXT-records and creates the renaming faces accordingly. The newly created named faces are used to rename Interests until they time out due to inactivity or outlive the TTL stated in DNS.

C. ContentObject Signature Validation

By renaming the ContentObject (CO) resulting from a successfully delivered Interest, one invalidates its signature as the signature is partially based on the name. Although the CO regains its validity when it is renamed to the original name at the initiating node, any node along the path may check its validity and drop a CO if it is invalid [24].

In order to prevent unjustified dropped COs two possible solutions exist. Currently, we encapsulate the original (signed) CO in a new CO with the renamed name and sign accordingly. Using this method the traveling ContentObject lives through signature checks at any point. The encapsulation of a CO in another CO however renders overhead. A solution containing less data overhead is to remove signature validation across the path and make the named faces and connected applications responsible for validation of the received ContentObjects. To support this feature, we propose to add a new type of flag to the Interest messages indicating whether the resulting ContentObject's signature may or may not be validated during transport.

D. CCNx DNS implementation

Since a mapping stored in DNS is a form of content itself, we can use the NDN to efficiently distribute the mappings among interested parties. In order to make DNS suitable for NDN, we have defined an application specific naming-scheme which describes the DNS request. The name contains

1) the authority hosting the DNS server: in order to forward the packages to the right party,

[9] If one or more location-aggregated names for a context-related name do not respond, the strategy layer will simply disregard those.

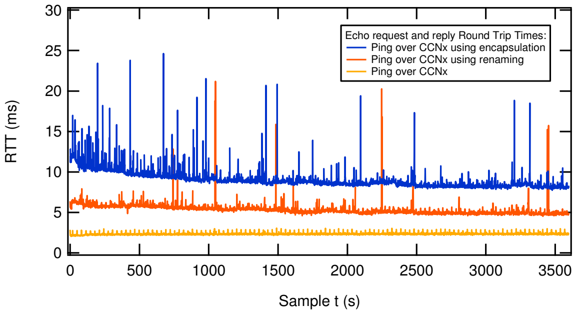


Fig. 3: Measured round-trip-times using Ping echo requests and replies over CCNx and renaming CCNx networks.

2) the name one wishes to query: the first NameComponent of the context-related name (e.g. Alice.eu), and
3) (optionally) the type of record one needs (e.g. A-, AAAA-, MX- or NS-records).

The NDN-record for the domain Alice.eu can be obtained by querying a local (recursive) DNS-server using the name /server.local/dns/Alice.eu/NDN and receiving a result from that server over the NDN. Using our CCNx DNS client and server extensions [25] one can extend existing DNS servers and clients with NDN capabilities. Once DNS root and Top Level Domain servers are also configured to support NDN, the NDN can be used to query all recursion steps.

E. Experimental Measurements

In order to verify the functionality of our renaming scheme, we have set up a small network containing 4 nodes connected in a straight line giving a diameter of 3 edges. In this network, we have set up experiments measuring the round-trip-time (RTT) between the end-hosts using 3600 consequent echo requests, submitting 1 echo request each second, and their subsequent replies. We have used *ccnping* [26] to confirm RTTs over the plainly routable and renaming NDN network. Although the network is rather small, the largest additional delay is generated by the two renaming nodes independent of the size of the network.

Figure 3 shows the RTTs for forwarding the packets using (1) a plain CCNx network, (2) a renaming CCNx network not considering signature validity and (3) a renaming CCNx network that guarantees valid signatures by encapsulating signed ContentObjects into new. The RTT averages of respectively 2.338, 5.327 and 9.031 ms show that the delay overhead by applying renaming in a network equals 2.989 ms while encapsulating and signing packets adds another 3.704 ms.

We consider the average delay of 2.989 ms to be a feasible penalty compared to the recurring profits of the decreased routing table sizes. Furthermore, we expect future optimizations [10][11] on comparing content names and FIB lookups to further decrease this delay. Finally, the additional transmission delay introduced by encapsulating packets supports our proposal from subsection V-C to disable intermediate signature validation for renamed packets.

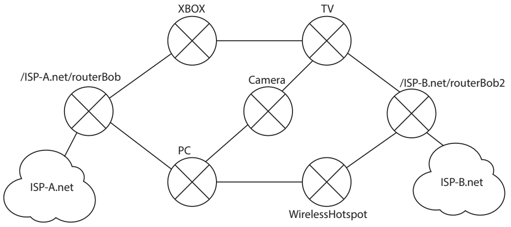


Fig. 4: An example LAN, in which the photos on the camera can be shared by both the names /ISP-A.net/routerBob/PC/Camera and /ISP-B.net/routerBob2/TV/Camera.

VI. Local Area Network Implementation

In this section, we will discuss the details of our LAN prototype implementation [27] using recursive name aggregation. As discussed in section IV, we propose to use the names of the nodes connecting a LAN to its ISP and use subnames of these names to base naming on within the network. This enables LANs and their names to form dynamically without the administration of reserving ranges and setting up forwarding manually. The shortest paths to all edge routers can be used as default forwarding rules between which the NDN strategy layer employs heuristics to benefit from the multipath routing rules. The reverse path vector of hostnames, added to the entrypoint's base name, dynamically generates a globally unique routable name for each node in the LAN as shown in figure 4. Where intra- and inter-AS routing delivers Interests meant for the LAN to the LAN edge routers, the routers within the LAN set up correct rules to forward Interests internally.

A. Propagation

In order to fulfill the propagation of available routes and names, our prototype routing discovery application [27] employs a path vector routing mechanism exchanging messages containing the following properties for each possible route:

- The originating LAN edge router, called entrypoint, to identify multiple entrypoints and multipath routes.
- The route one can offer: either a gateway route, a specific route or the aggregation base name (to ensure one can reach all nodes within its own LAN).
- The cumulative cost of the offered route.
- The path vector to the entrypoint, which can be used for both naming and loop detection.
- A Boolean indicating whether aggregation upon the offered route is permitted.

When a node enters the network, it broadcasts a discovery message on all its interfaces indicating that it is new and needs to receive all possible routes neighboring nodes are willing to forward. It receives a set of routes containing the previously discussed properties from each neighboring node. From the received sets, the new node selects the optimal subset of routes it will use based on cost and path vector, including multipath routes that are identified based on the originating entrypoint. The node requests the usage of the selected routes from the neighbors in question, which will set up the necessary forwarding and acknowledge thereafter.

At this point the new peer has established its state within the network and may offer forwarding and aggregated names to even newer peers recursively. The end-user can start sharing content globally using the received location-aggregated names and may register the names at a mapping service as discussed in section V to publish information using context-related names.

VII. Related Work

The idea of name aggregation was first proposed in a technical report [2], where it is referred to as ISP-based aggregation. In this paper we have further enhanced this aggregation by applying a recursive extension of names when one descends into the network, as well as proposing dynamic generation of those names in Local Area Networks. Where the report proposes to have the PC's themselves map "pretty" names into ISP-based names in an early phase, we propose to use a late binding of identifiers as implemented in the Locator/Identifier Separation Protocol [13].

The NDN name conventions [28] only state that names have to be hierarchical in nature. Stricter conventions are solely enforced by the applications or institutions using NDN. Only in the case of globally reachable names, a name should be globally unique and its first name component should be a "DNS name". Further name conventions regard markers such as version timestamps and segment numbers which do not influence routing as they are appended to the end of a name.

Since our proposal contains first name components that can be registered for both context-related as location-aggregated names[10], both can be inserted into the DNS system and used accordingly. Furthermore, [5] and [28] state that the DNS may be used for looking up IP tunnel endpoints, which can be used to bridge NDN incompatible nodes and ASes.

Finally, recent research has been done in the area of speeding up NDN table lookups [10][11].

VIII. Conclusion

In this paper, we have proposed to decouple context-related NDN names from routable names. By using routable names aggregating to Autonomous Systems and their internal structure, the size of global routing tables becomes upper bounded by the number of ASes in a system. By using a LISP-style late binding of context-related names to location-aggregated names using the DNS directory service, we enable applications to communicate based on context-related names. Our implementation of mapping and renaming context describing names to routable names, enables the NDN strategy layer to facilitate multipath routes and multihomed applications.

[10] Both domain names and ASNs already need to be registered at the appropriate administration organizations.

References

[1] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," *SIGCOMM*, 2007.

[2] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, K. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, "Named Data Networking (NDN) Project NDN-0001," Palo Alto Research Center, Tech. Rep., Oct. 2010.

[3] C. Dannewitz and T. Biermann, "Prototyping a network of information," *IEEE Local Computer Networks*, vol. 34, 2009.

[4] N. Fotiou, P. Nikander, D. Trossen, and G. Polyzos, "Developing information networking further: From PSIRP to PURSUIT," *BROADNETS*, Oct. 2010.

[5] V. Jacobson, D. K. Smetters, J. Thornton, M. F. Plass, N. Briggs, and R. Braynard, "Networking Named Content," *CoNEXT 2009*, 2009.

[6] "Internet grows to more than 240 million domain names in the second quarter of 2012," Verisign, Tech. Rep., Oct. 2012. [Online]. Available: https://investor.verisign.com/releasedetail.cfm?releaseid=710803

[7] "CIDR Report," CIDR, Tech. Rep., Dec. 2012. [Online]. Available: http://www.cidr-report.org/as2.0/

[8] R. A. Steenbergen and R. Mosher, "An inconvenient prefix: Is routing table pollution leading to global datacenter warming." NANOG50, Oct. 2010.

[9] L. Wang, A. K. M. M. Hoque, C. Yi, A. Alyyan, and B. Zhang, "OSPFN: An OSPF Based Routing Protocol for Named Data Networking," Palo Alto Reserch Centers, Tech. Rep., Jul. 2012.

[10] Y. Wang, K. He, H. Dai, W. Meng, J. Jiang, B. Liu, and Y. Chen, "Scalable name lookup in NDN using effective name component encoding," in *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, 2012.

[11] H. Yuan, T. Song, and P. Crowley, "Scalable NDN forwarding: Concepts, issues and principles," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, 2012.

[12] P. Mockapetris, "RFC 1034 domain names - concepts and facilities," Nov. 1987.

[13] D. Farinacci, "draft-ietf-lisp-23 Locator/ID Separation Protocol (LISP)," May 2012.

[14] D. Meyer and K. Patel, "RFC 4274 BGP-4 Protocol Analysis," Jan. 2006.

[15] T. Bates, Y. Rekhter, R. Chandra, and D. Katz, "RFC 2858 Multiprotocol Extension for BGP-4," Jun. 2000.

[16] RFC Editor, "RFC 5000 Internet Official Protocol Standards," May 2008.

[17] A. Narayanan, S. Previdi, and B. Field, "draft-narayanan-icnrg-bgp-uri-00 BGP advertisements for content URIs," Jul. 2012.

[18] J. D. Clercq, D. Ooms, S. Prevost, and F. L. Faucheur, "RFC 4798 Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)," Feb. 2007.

[19] G. Lutostanski and B. Zhang, "NDN-Routing/ccnx-dhcp," Dec. 2011. [Online]. Available: https://github.com/NDN-Routing/ccnx-dhcp

[20] R. Cox, A. Muthitacharoen, and R. Morris, "Serving DNS using a peer-to-peer lookup service," *Peer-to-Peer Systems*, pp. 155–165, 2002.

[21] L. Jakab, A. Cabellos-Aparicio, F. Coras, D. Saucez, and O. Bonaventure, "LISP-TREE: A DNS Hierarchy to Support the LISP Mapping System," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1332 – 1343, Oct. 2010.

[22] V. Fuller and G. Wiley, "draft-fuller-lisp-ddt-04 LISP Delegated Database Tree," Sep. 2012.

[23] N. L. M. van Adrichem, "CCNx-RenamingFaces," Dec. 2012. [Online]. Available: https://github.com/TUDelftNAS/CCNx-RenamingFaces

[24] Palo Alto Research Center, "CCNx Protocol," Mar. 2012. [Online]. Available: www.ccnx.org/releases/latest/doc/technical/CCNxProtocol.html

[25] N. L. M. van Adrichem, "CCNx-RenamingFaces," Dec. 2012. [Online]. Available: https://github.com/TUDelftNAS/CCNx-DNSRelay

[26] C. Yi, "NDN-Routing/ccnping," Jan. 2012. [Online]. Available: https://github.com/NDN-Routing/ccnping

[27] N. L. M. van Adrichem, "CCNx-DHCNGP," Dec. 2012. [Online]. Available: https://github.com/TUDelftNAS/CCNx-DHCNGP

[28] Palo Alto Research Center, "CCNx Basic Name Conventions," Feb. 2012. [Online]. Available: http://www.ccnx.org/releases/latest/doc/technical/NameConventions.html