MNDN: Scalable Mobility Support in Named Data Networking

Xavier Mwangi
Massachusetts Institute of Technology
Cambridge, MA
xmwangi@csail.mit.edu

Karen Sollins
Massachusetts Institute of Technology
Cambridge, MA
sollins@csail.mit.edu

ABSTRACT

In this paper, we present MNDN (Mobile NDN), a scalable design for mobility support of producers in the Named Data Network (NDN) architecture. The approach in this work is to separate concerns of scalability and mobility, noting that there are two aspects to mobility: 1) discovery that a part of the namespace can be mobile; 2) discovery of where that part of the namespace is currently attached to the network. This separation leads to the observation that the information for these two aspects have different performance requirements, driving us to a partitioned design, basing the scalability component of the design on NDN's SNAMP and the mobility support on MobilityFirst's DMap. The combination of these two components yields a mobility solution that allows NDN's forwarding to operate and scale in the face of mobile clients and producers.

CCS CONCEPTS

• **Networks** → **Network architectures**; *Network services*; *Network management*;

KEYWORDS

Named Data Networking, mobility, scalability

ACM Reference Format:
Xavier Mwangi and Karen Sollins. 2018. MNDN: Scalable Mobility Support in Named Data Networking. In *5th ACM Conference on Information-Centric Networking (ICN '18), September 21–23, 2018, Boston, MA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3267955.3267970

1 INTRODUCTION

In contemporary networked environments, we find two significant demands: scalability and support for mobility. As devices become smaller and more plentiful, the sources and destinations of traffic grow. In addition, the devices themselves are increasingly mobile. For our work here we focus on scalability in the form of reachability. The mobility that interests us may be reflected as movement of attachment point within a protocol layer and stack or movement between technologies. Each individual element may be moving frequently, and the totality of mobility events is exploding.

ICN '18, September 21–23, 2018, Boston, MA, USA

ACM ISBN 978-1-4503-5959-7/18/09…$15.00
https://doi.org/10.1145/3267955.3267970

In this context, we also see a proposed revolution on communications semantics from byte-stream or flow based communication between end-points, to information exchange, as proposed in NDN [10], based on retrieving named content. The question we address in this paper is how to provide simultaneously scalable content retrieval when the producers of the content are mobile. The challenge is to make both scalability and particularly mobility seamless, so that the client does not experience significant delays, especially during mobility events. The original design of NDN considered client mobility, but not producer mobility.

The approach we take in designing Mobile NDN (MNDN) is a separation of functions. In order to provide scalability in reachability, the well-understood approach is to separate names with well-known routing resolution from names that must be forwarded to another named authority for resolution. The keys to our design are (1) distinguishing those name prefixes that are homed by potentially mobile producers; (2) a simple, efficient protocol for signaling a mobility event; and (3) a separate scalable, fast, pervasive service for tracking mobile events. For the first of these capabilities we choose the Secure Namespace Mapping service of NDN [4], modified to include special records for names that are potentially mobile. For the second, we use a simple negative acknowledgement protocol. And, for the third, as a proof of concept we use the name mapping service of MobilityFirst, DMap [23]. This separation of services reflects the different performance requirements of standard, non-mobile routing and of the type of mobility we believe will be pervasive.

The paper will proceed as follows. Section 2 begins with an expansion of our approach to both scalability and mobility. In Section 3, we review related work, including brief reviews of the NDN and MobilityFirst architectures as needed here. We discuss the system design, justification for that design, and components in Section 4, addressing how *time to detect* and *time to respond* to mobility events are handled more effectively separately. In Section 5, we reflect on choices for mobility support in the larger ICN community, security, and garbage collection. In Section 6 we evaluate the design briefly. See Mwangi [17] for greater detail. The paper concludes with a discussion of its contributions and future work.

2 THE PROBLEM

The challenge of providing scalability and time-sensitive mobility for data providers in NDN hinges on the fact that NDN is intended to hide rather than expose location information. In NDN, the information that reflects how to reach the authoritative or published version of named content resides and is managed through a FIB update protocol, derived from existing Internet routing protocols. At scale and at distance these are well-understood to be slow, in part in order to damp out local fluctuations. In order for an interest packet to reach a moving producer, we need an alternative plan.

The problem is that because there may be many mobile producers each responsible for a subhierarchy of the global naming hierarchy there may be explosive growth in mobility events, yet access to any data should be accessible, both efficiently and quickly.

The first part of the problem is how to recognize and keep track of those points in the hierarchy where mobility is occurring. For that we depend on a name service that describes how to reach a moved publisher in the short term, by way of indirection to a non-mobile attachment point. The second part of the problem is to signal to the client that the producer of the data for which it has expressed interest has recently moved. The third is to provide the most recent attachment information. The interesting observation is that in the longer term, the normal NDN FIB update protocol will catch up and stabilize.

The challenge that this paper addresses is to provide seamless fast support for mobility. In Section 3, we will review the existing approaches and specifically the concepts from prior work on which we will base our own design. Section 4 will then present that design.

3 RELATED WORK

At the core of the work related to this project are the questions of name structure and resolution. We will begin with NDN's integrated model and then briefly identify why and how one might separate the different functions provided by name resolution in order to address the name/location problem that arises from support of producer mobility discussed above. Because our approach combines aspects of naming and name resolution from both the NDN [10] and MobilityFirst [18, 22] projects, both will be touched on.

NDN in its original and simplest form proposed a model for retrieval of content with hierarchical names, through an *Interest* packet, to which a *Data* packet is the response. To enable this, every node in the network would have a forwarding information base (FIB) managed by a FIB update protocol, and a pending interest table (PIT). In addition, it was likely to have a content store for caching data packets by name. To deliver an interest packet, the FIB first performs longest prefix matching on the NDN name to determine the appropriate outgoing interface. It then forwards the packet, leaving within the PITs a trail of breadcrumbs that are used to route the requested Data packet. Caching via the content store is used to shorten retrieval times. We note that location information is hidden in the FIBs along the path, and not represented explicitly anywhere.

Stepping back, if we consider the current IP network, the separation of hierarchical DNS names [14, 15] from IP addresses makes explicit the idea of location. In fact, we actually use IP addresses for both location, as in routing, and identification, as in identification of TCP flows. Two notable works to separate these are the Host Identifier protocol (HIP) [16] and the Identity-Locator Network Protocol [5], both of which make the distinction between identity and location explicit. Our approach provides a similar separation of name and location, starting with the Secure Namespace Mapping service (SNAMP) [4].

Finally, we note, because it is important to our approach, the MobilityFirst project also proposes a separation of these concepts. They propose a name resolution service layer that maps user friendly names to Global Unique Identifiers (GUIDs), which in turn identify network elements. The Global Name Resolution Service (GNRS), of which DMap [23] is one version (the others [8, 20] discussed in Section 5), maps GUIDs to network addresses, the routable identifiers of local regions of the network, similar to autonomous systems (ASs). It is the GNRS that is designed explicitly to handle location mobility at scale and frequency not supported by any of the other resolution services in MobilityFirst. Our work builds on this service design.

In order to address producer mobility in the context of NDN, there are two more related pieces of work important to us here. First we return to SNAMP [4]. This work addresses the fact that some NDN names may not be routable. Afanasyev et al. propose that there will be default-free zones (DFZ),[1] which are routable, to which they will forward the non-routable names using DNS for NDN (NDNS) [2] delegation. SNAMP defines *link objects*, to be placed in interest packets, that contain routable hints to the region where the non-routable name can be resolved. In our work, we use a link object to contain the new attachment location after a mobility event. It is retrieved by fast lookup in our GNRS after a NACK signals that the producer has moved.

The final reference that it is important to note here is Zhang et al.'s survey of mobility approaches in NDN [26], because it is useful to recognize that our approach falls into their taxonomy as a *mapping* type service with *interests with hints*, because it includes features of both SNAMP and DMap.

4 DESIGN OF MNDN

In this section, we present an overview of our system design followed by a discussion of the basis for that design, and then examine each of the components separately, including mobile zones of names, an external link store based on DMap, the security of link objects, caching, and an overview of data retrieval in MNDN.

4.1 Overview

MNDN builds on the ideas in SNAMP and draws insights from the MobilityFirst project to achieve our seamless mobility goal at scale. In order to minimize data retrieval delay in the face of mobility, we present approaches to minimize *time to detect* that a mobility event has occurred and the *time to respond* to a mobility event. MNDN:

(1) Defines and makes discoverable the names that may be mobile, and
(2) Quickly maps data names to routable prefixes.

This decoupling comes from the observation that the information required to support each of these two mechanisms has different lifetimes. In general, whether data exists on a mobile producer is long-lived information, while the location of the mobile producer is short-lived information.

MNDN defines which name prefixes may move and uses NDNS to maintain and serve this information. Given a data name, MNDN can determine the name prefix that may have moved and consults a Global Name Resolution Service (GNRS) to receive the current reachable prefix of the moved name prefix. Consider the example in Figure 1. To ensure seamless mobility, when the mobile producer

[1] A default-free zone in the IP network is composed of BGP routers that have complete routing tables with no default routing entries. The term was adopted in SNAMP to have a similar meaning.

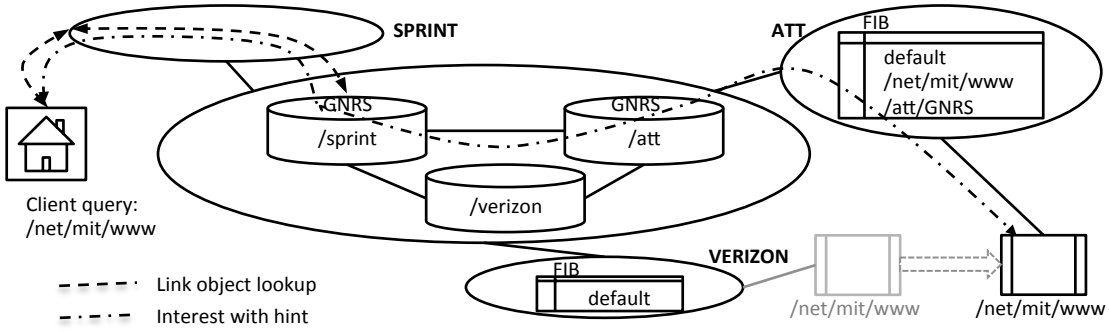

Figure 1: Mobility Scenario.

of the prefix `/net/mit/www` moves from the `/verizon` network to the `/att` network, several steps must take place:

(1) The mobile producer updates the GNRS with its new globally reachable prefix;
(2) The client in the `/sprint` network quickly detects that the data prefix; `/net/mit/www` has become unreachable
(3) The client determines which of the three possible prefixes has become unreachable (i.e. `/net/mit/www` or `/net/mit` or `/net`);
(4) The client queries the GNRS for the current reachable prefix of the moved prefix;
(5) The client attaches the globally reachable prefix to the interest packet for `/net/mit/www` and sends it out again.

4.2 Separation of Concerns in Mapping

As mentioned above, although SNAMP was originally designed to handle both scalability and provider mobility to some degree, we find that it is inadequate for our model of mobility. To address these problems, MNDN relies on separation of concerns at two levels.

First, instead of designing a single mapping function to solve both namespace scaling and mobility, we extend SNAMP with additional mapping functionality. As a result MNDN is composed of two relatively modular mapping functions that operate to achieve mobility at scale. Second, we separate the problems of identifying mobile data and locating mobile data, because the lifetimes of that information is different. This separation has the following benefits:

- There is reduced overhead in initial object lookup when compared to two independent mapping systems. When retrieving the link object associated with a mobile zone, the client can rely on cached information from NDNS to determine that a direct lookup into the GNRS is required. In addition, we have fast failure because an NDNS query for the link object associated with a non-mobile zone doesn't initiate a lookup into the GNRS.
- The existence of mobile link objects decreases the size of the namespace that the GNRS must manage and decreases the query traffic that the GNRS must serve. Effectively, NDNS is used as a filter such that the domain of the GNRS is only those prefixes that are known to be mobile.

With these design decisions in mind, one can take MNDN as describing a family of mobility solutions that address namespace scaling by extending SNAMP. The choice of DMap is taken as a proof of concept, but any non-tunneling mapping solution should be compatible with the described framework. However, the advantages of separating mobile data identification and location is only beneficial for mapping-based mobility solutions. Mobility solutions that rely on the forwarding plane may extend SNAMP with limited concerns about NDNS but will have to consider the effects of DFZs and link object-based forwarding.

We observe, however, that the first concern led to a separation of functionality, and the second to strengthening the link between them.

4.3 Mobile Zones

While human-readable hierarchical names for content are generally desirable, their use in NDN introduces complexity when considering producer mobility. Consider the example in Figure 1 where the globally routable prefix associated with the name prefix "`/net/mit/www`" changes from "`/verizon`" to "`/att`". In SNAMP's data retrieval model, when the client sends an interest packet containing a stale link object, it will receive a `NACK` reply. This indicates that, even upon reaching a router for which "`/verizon`" is local, the interest could not be forwarded further. However, at this point, it is unclear which prefix became unreachable. Is it "`/net`" or "`/net/mit`" or "`/net/mit/www`"? With this ambiguity unresolved, the mapping system would have quickly to determine the globally reachable prefix for each of these name prefixes.

To resolve this ambiguity, in MNDN, a client must know, apriori, which prefixes can move. To accomplish this, MNDN introduces *mobile zones* to NDNS. A mobile zone is contiguous a portion of the NDNS namespace that is served by at least one mobile device. Conceptually, mobile zones are exactly the namespace prefixes that a mobile producer advertises. That is, during the process by which a mobile node acquires the authority to publish under a namespace prefix, that namespace prefix becomes a mobile zone. Given that SNAMP does not support rapid updates of link objects that seamless mobility requires, mobile zones may be unreachable by relying on FIB and NDNS information alone. However, the fact that a zone is mobile is information that can be stored in NDNS.

Using NDNS, a nameserver that is authoritative for a zone can report that children zones are mobile. We define a new NDNS record type, `MOBILE`, to hold mobile zone information. At a minimum, the

MOBILE record contains a boolean value that indicates whether a zone is mobile. In our example, when "/net/mit/www" becomes a mobile zone, a MOBILE record is created and stored in the parent zone's nameserver, usually reachable by "/net/mit/NDNS". Mobile zone discovery then proceeds, as in DNS, through an iterative resolution process. Note that, since this record is considered long-lived, it can be cached for an extended period by caching resolvers.

4.4 External Link Object Store

Mobile zones and non-mobile zones require a mapping system with different properties. As MNDN can identify which zones are mobile, it can handle the retrieval of link objects associated with mobile zones differently. Such link objects change frequently and, therefore, the mapping system that maintains them must support rapid retrieval and update of link objects. In the rest of this paper, we call link objects associated with mobile zones *mobile link objects*.

There are two general approaches to storing mobile link objects. The first approach mirrors that of NDNS and requires each zone to maintain a mobile link object store. This approach has some challenges. The frequent updates mean we can not rely on caching and would require each query to travel to the zone's local mobile link object store. Such an approach can not meet our latency requirements as the zone's object store might be located far from the client. This suggests that each mobile zone must maintain a globally distributed mobile link object store with sufficient replication to support rapid responses to queries.

For the global mobile link object store, we propose to use DMAP [23] from MobilityFirst, because of its fast update, fast lookup and wide availability to resolve mobile zone locations in NDN. For simplicity, we assume that each prefix in the DFZ is associated with a single autonomous system (AS). As in DMap, we store the map from mobile zone prefix to globally reachable prefix within the network routers. Given a mobile zone prefix, we use K consistent hashing functions to generate K random values, as is done in the original DMap design. These random values are mapped uniformly to the known DFZ prefixes, and the mobile zone mapping stored within the routers of the corresponding AS. [2] To accomplish this, we have the gateway routers of each AS advertise the names "/GNRS" and "/GNRS/<DFZ-prefix>". Note that in this design, each gateway router maintains a list of all DFZ prefixes in order to perform the uniform map from random number to DFZ prefix. This is reasonable as the gateway routers necessarily maintain this information in their FIB and can all establish a consistent ordering using lexicographical sort.

In the process of link object resolution using NDNS, if a client (or recursive resolver) receives a MOBILE record as a reply, it must query the GNRS for the link object instead. The client sends the query to its local GNRS (found at "/GNRS") which performs the hashing function and forwards the query to one of the K GNRS responsible for the mobile zone.

4.5 Secure Delegation of Link Objects

As the designers of SNAMP noted, all critical information must be signed. MNDN builds on the hierarchical trust model used by NDNS. In the NDNS model as in the DNS, all zone records are signed by at least one *data signing key* (DSK), and DSKs are signed by at least one *key signing key* (KSK). The KSKs are, in turn, self-signed (for well-known zones) or signed by *delegation keys* (D-KEY) stored in the parent zone. Our design directly inherits certification of the MOBILE record stored in NDNS.

The mobile link objects discussed above allow us to address the following threats:

(1) **An attacker may map a zone she doesn't own to a DFZ**: We can consider each mobile device that owns a mobile zone to extend NDNS. When a mobile device becomes the owner of a mobile zone, it generates a DSK that is used to sign the mobile link objects it places in the mobile-link object store. These DSKs are then signed by the zone's KSK.
(2) **An attacker may map a zone she owns to a DFZ without the DFZ's knowledge**: We must certify that the DFZ we attach to is aware that we are routing traffic towards it. We assume there is a process by which the mobile device acquires permission to route data towards the DFZ. If our scheme involved appending link objects to the data name, this would be analogous to the process acquiring permission to advertise under the DFZ's prefix. During this process, the DFZ signs the mobile device's DSK with a KSK. For simplicity, we can assume that DFZs are well known and can provide self-signed KSKs. If, not, however, we can rely on the hierarchical trust provided by NDNS once again.
(3) **An attacker may modify link object in-flight to and from the link object storage**: The integrity of the data in-transit is guaranteed by having the mobile-link object signed by the mobile device's DSK.
(4) **An attacker may unmap a zone she doesn't own**: This effectively mounts a DoS attack on the mobile zone. We require each mobile-link store to verify an update before updating its records.

4.6 Caching

Some care must be taken in introducing caching to MNDN as it brings along the possibility of stale information. After all, it is caching that makes NDNS queries fast while simultaneously leading to slower updates when compared to DMap. We present two cases where caching can be used safely in MNDN and to great benefit.

First of all, MNDN does not allow the caching of mobile link objects by caching resolvers – they are only allowed to cache MOBILE records. However, we do not disallow caching of mobile link objects by the client. As a result, retrieving data served by a mobile producer does not require queries to the GNRS unless the mobile producer moves. This follows from the fact that MNDN only activates in response to a NACK. During normal operation, the mobile link object, as would a standard link object, is reused by the client for retrieving data within a zone. When the client receives a NACK, it knows that its local cached copy of the mobile link object is stale and performs a lookup in the GNRS.

Secondly, we introduce the use of mobile link objects as mobility signals. As it stands, for the client to detect that a mobile producer has moved, it must first send out an interest packet containing a

[2] The k hash functions are the basis for selecting replica servers with DMap, in order to acheive even, random distribution.

stale link object and then receive a NACK. In general, this process may incur non-negligible cost: the original attachment point of the mobile producer may be far and the lifetime of the packet can lead to a significant wait before the NACK is produced. To alleviate this issue, we allow for link objects and the time of their retrieval to be cached in routers. To this end, we modify mobile link objects to include a timestamp of their creation time. When a router processes an interest packet with a link object, it also checks whether it has a link object for the same mobile zone. If the router has a newer link object, it returns a NACK to the client. If it does not have a link object for the mobile zone, or the link object it has is older than that in the link object in the interest packet, it forwards the interest packet as it normally would. Overall, this allows clients to learn that a mobile producer has moved based on other clients having recently queried the GNRS for this information earlier. This mechanism has similar characteristics to the caching of standard data within routers. That is, it is ultimately more effective for popular data and compounded by the effects of temporal and spacial locality of data requests.

4.7 Retrieving Data in MNDN

The process of retrieving data in MNDN differentiates between mobile zones and non-mobile zones. When a zone is not mobile, data retrieval takes place as described in SNAMP. If an interest carries a name that is not in the DFZ and can not be satisfied by a cached copy along the way, MNDN takes over. A default-free router will reply to this interest packet with a NACK. This indicates that, to forward the packet further, the router requires additional information in the form of a link object. MNDN, using the NDNS, retrieves and verifies the link object. If the zone is mobile, we discover this information as the NDNS resolution takes place, and delegate part of the resolution to the GNRS. After the link object is retrieved, the client will place it into the original interest packet and send it out. When this modified interest packet reaches a router that is unable to forward using the interest name, it forwards by selecting the best among the prefix delegations in the link object. On subsequent NDNS resolutions for this name, we retrieve a cached copy of the MOBILE record which indicates that we should query the GNRS for the link object.

There are two points that we we must address. First, a word is in order regarding names in the DFZ. If a prefix such as /att/www moves, an interest for data within this prefix will result in a NACK. However, standard SNAMP does not require that such names be placed in NDNS. In MNDN, we require that upon becoming a mobile zone, prefixes within the DFZ are also placed in NDNS and by, extension, the GRNS. Second, we may also come across a situation where FIB information is outdated and conflicts with link object information. Such a situation may occur when the client and mobile producer are initially in the same network, but the mobile producer moves, leading to temporary looping. However, interest lifetime guarantees that eventually the client receives a signal that the interest has expired. We treat this signal in the same way as we do a NACK and rely on the FIB update protocol to guarantee that this transient state is short-lived.

5 DISCUSSION

In this section, we briefly address four key topics with the above design: (1) separation of concerns in mapping; (2) several security issues; (3) the question of nested mobile zones; and (4) garbage collection.

5.1 Choices in Designing for Mobility

In this work, we have proposed a partitioned solution to achieve scalability and mobility in concert. We have taken as given that SNAMP is the best solution for scalability in NDN. But we are left asking: (1) Would a single unified service be more effective? (2) If not, is there an alternative design for mobility that would be preferable?

To address the first question, assuming SNAMP, one is asking whether integrating some improved mobility service into SNAMP would be reasonable, because it has the beginnings of a mobility service that is based on NDNS. The key issue with the SNAMP approach is that the NDNS is not seamless in terms of resolution, at rates and scales of content retrieval, nor is its update scheme capable of handling the quantity of update requests to make mobility events trackable in real time. Thus, we propose to use NDNS within SNAMP to provide the slower changing and more cachable information about which name prefixes are potentially mobile, but not their actual mobility.

Thus, assuming that a separate service will better serve our needs, we turn to the survey of NDN mobility approaches Zhang et al. [26] and further work on MAP-Me by Auge et al. [6] To summarize, Zhang et al. define three types of mobility support services, *mapping*, *tracing*, and *data storage*. Simply put, mapping services translate names into something else, to be returned to the client for re-request through a new interest. Tracing services track the mobile producer by forwarding the interest packet from the original location through a trace produced by the mobile producer to the current location. Data storage services provide static storage locations for content that would otherwise be found on the mobile producer. Auge et al. further refined the tracing approaches by recognizing that some have anchors, stable locations from which all tracing emanates for a particular mobile producer and some that are anchor-less. In fact, their solution is anchor-less.

Let us work backwards through the alternatives. The problem we face with remote stable storage as in [7, 9] is that it is not general purpose with respect to the nature of the data. Only stable, long-lived data can be handled reasonably by such services. Streaming and other dynamic data will only incur significant performance degradation. Tracing solutions found in [6, 11, 24] have a different set of problems. The information about mobility is not wide-spread, so all interests sourced outside the range of the route of the tracing information will experience path-stretch until the slow FIB update protocol distributes the new location information. This is true for both anchored and anchor-less tracing. This leaves us with the set of mapping services, of which SNAMP itself is a leading contender. We have already concluded that SNAMP on its own is not adequately responsive to large numbers of fast moving mobile producers, with respect to their immediate location information. Therefore, in this work, we have proposed our DMap solution as an ancillary service to SNAMP as a prototypical widely distributed,

highly responsive mapping service, to address the performance requirements we envision.

5.2 Security Issues

Although security is not central to our design, there are a number of security issues that we address here. The first is the general question of how secure are the components we are composing into a service. For this we briefly review the security architectures defined for SNAMP with NDN and DMap. We recognized that there remain several further questions; we focus on two here, link object security, since link objects may be corrupted outside the services, and denial of service attacks. We acknowledge that we do not have solutions to these, but recognize that they deserve further thought and attention.

We begin with the work of Tourani et al [21]. As a starting point, NDN signs the combination of data and its name. It also provides certificates by signing names with their public keys. In addition, NDN provides a trust framework including trust schemata and trust anchors based on SDSI [19]. This basis means that names can be mapped to content in a trustworthy way, ownership, responsibility and integrity can be tracked, policies enforced, and a hierarchy of trust developed. On top of this, NDN also defines the Secure Namespace Mapping (SNAMP) [4] based on the *Map-and-Encap* idea to achieve secure scalable management of forwarding information. This all provides for authentication, authorization, integrity verification within the services, and policy management.

We then compose that with DMap, for which we turn to the work of Liu et al. [13] In this work, the authors design, build, and demonstration protocol extensions including for elimination of: (1) unauthorized resource consumption in various forms, (2) unauthorized use of services, and (3) a variety of denial of service attacks. To complement that Liu et al. [12] specify and include an attribute-based-access control (ABAC) policy language based on XACML. Thus, DMap's security architecture provides access control, denial of service (of the service itself) and a strong policy control language for specification.

Because these two services are operating independently, they do not need to interoperate in order to achieve their own security attributes, but that does not eliminate security issues between them and among clients and producers in NDN. We consider two potential areas of concern, security issues with link objects themselves, as they exist in NDN, and possible denial of service attacks in the interaction between SNAMP and DMap.

SNAMP considers security carefully and observes that, even with signed link objects, the possibility of cache pollution remains. By modifying the link object within an interest packet, a malicious router can redirect the interest towards a malicious producer. This malicious producer generates incorrect data which gets cached along the routers along the path traversed by the interest packet. While this threat can be resolved by having hop-by-hop verification of the signed link object, this approach is prohibitively computationally expensive. SNAMP chooses to cache this possibly incorrect data but to identify it using a name and link object pair instead of by the name alone. As a result, requests for the same data name but containing a different link object can not be satisfied by the cached copy. While effective in a low-mobility scenario, this security model presents some challenges.

First of all, the original (unsecured) SNAMP design [3], considered the ability for in-network routers to modify the link object within an interest packet to various effects. However, this can not be accomplished in MNDN; we cannot, without excessive cost, allow link objects to be modified safely, because the discovered link object is embedded in the interest packet and is used by all subsequent routers.

Furthermore, the use of a name and link-object pair to identify data in caches decreases the effectiveness of the cache. When the link object associated to some mobile zone changes, a client will transition to requesting data within the mobile zone using this updated link object. While using the stale link object, caches can satisfy interest packets. If the mobile producer is not constantly moving, the caches will be eventually repopulated with data identified using the updated link object. However, in the future to accommodate constantly moving producers we may consider allowing the possibility of invalid data in caches and require that clients perform verification. Ultimately, both approaches waste space in the cache, in the form of invalid data or unusable data.

Our final security concern here is with denial of service, in particular, the possibility that DMap may become inaccessible, making mobile producers themselves inaccessible. We do not have a specific solution here to DoS floods, but note two features. First, DMap is widely distributed with significant replication of data. Therefore a DoS attack on any single server or path to a server will not bring the service down. Second, the use of DMap is for temporary data. It is designed to provide short term information about reachability of a recently moved producer. The longer term FIB update protocol will provide FIB updates in the longer run. Thus, a temporary outage with respect to DMap will not cause the overall system to fail, but only to be slower and less accessible temporarily. In the long run, we urge further work on reduction in DoS vulnerabilities.

5.3 Garbage Collection

We can leverage the mobility patterns of human-owned devices to decrease the working size of the GNRS. In particular, evidence shows that mobile devices spend most of their time in a few locations [25] – their "home" networks. Within NDNS, we can store link objects that map mobile zones to the reachable prefix of their home network. When a mobile producer moves to one of these home networks, it updates the GNRS to clear the mobile delegation information stored within. As a result, when a client wants to retrieve data from a mobile producer which is at home, the GNRS query returns a `NACK` which lets the client know that the delegation information in NDNS is usable.

6 EVALUATION

In this section we evaluate performance and overhead of MNDN. We first use a comparison to MobilityFirst's operation under mobility to demonstrate the effectiveness of our mobility solution. Simulation studies show that the use of mobility signals decreases the time to detect that a mobility event occurred. Lastly, we analyze the storage and system overhead that MNDN introduces and investigate how this load is distributed among the various autonomous systems.

6.1 Data Retrieval Latency

We provide a comparative evaluation of the mobility solution in MNDN to that provided by MobilityFirst. To compare these two systems we use the idea of a *network transit*, the delivery of a packet which provides no latency guarantees. We contrast network transits with GNRS queries, the latter whose latency is generally bounded.

Consider the scenario, in both MNDN and MobilityFirst, where upon sending a packet that represents a request for data, the mobile producer moves. We ignore the startup costs of initially discovering the location of the mobile producer. In the process of data retrieval, MNDN incurs four network transits and one GNRS query. Two network transits arise in expressing an interest with a stale link object, and from the returned NACK. At this point the client makes a GNRS query. Two additional network transits arise from re-expressing the interest with an updated link object and from the retrieval of the requested data. Notice that even if we could have MNDN routers perform the GNRS query and modify the link object (see Section 5.2), we would still effectively require four network transits due to path stretch. In contrast, MobilityFirst incurs the cost of three network transits and two GNRS queries.

Although we are unable to bound network transit latency, we have some control over GNRS performance. With replication of link objects in five ASs (K = 5), DMap has been shown to achieve latencies within 100ms [23]. While GMap [8] and Auspice [20] achieve lower query latency, their reliance on geolocation makes them less natural in NDN. Overall, compared to standard operation, MNDN introduces the additional latency of two network transits and a GNRS lookup. In effect, this a very minor disruption in network operation that is indistinguishable from the infrequent packet loss observed in networks.

6.2 Mobility Signals

We introduced mobility signals (see 4.6) to decrease the time to detect that a mobility even has occurred. We generated data traces by modeling data interests according to a Zipf distribution and assumed a uniform probability of mobility. The simulation was conducted on a generated topology of 2000 ASs with an average 100ms latency between connections [17]. The goal of this change is to decrease the cost of the two network transits that arise from expressing an interest with a stale link object. We see in Figure 2, that mobility signals decrease the time to detect a mobility event quite significantly. In particular, focusing on the low-latency regime, mobility signals allow for the NACK to be returned in under 100ms for over 20 percent of interests sent with stale packets. Without mobility signals, less than 5 percent of these interest packets can receive a NACK in under 100ms.

6.3 Overhead and Load Distribution

To evaluate the overhead of MNDN, we first provide an estimate of the mobile link object size. In this analysis, we do not consider the cost of the discussed security mechanism (Section 4.5), but needless to say, the inclusion of signatures and appropriately sized keys would dominate. For simplicity, we assume that names are ASCII encoded and consist of at most 10 components, each of at most length 30 characters. This can be encoded in approximately 2480 bits. To support multi-homing, we assume a maximum of four

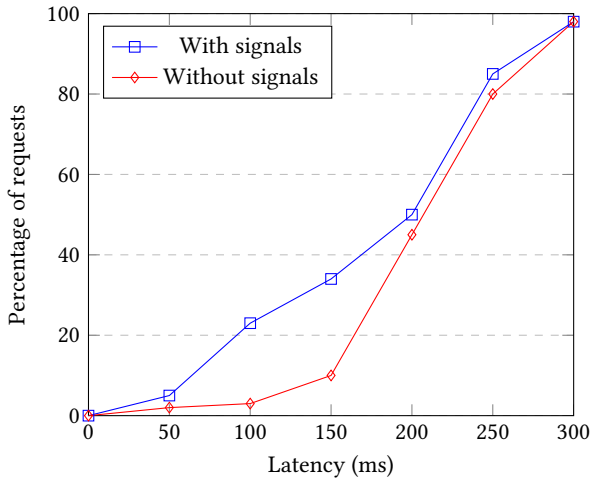

Figure 2: Signaling Performance

delegations per mobile zone. In total, each mobile link object will require ~1.5KB.

To estimate the total capacity required by the GNRS and load per AS, we assume a total of 10 billion mobile devices each advertising 10 mobile prefixes. In this use case, the total size of data in the GNRS comes out to ~150TB. If we assume NDN at internet scale is organized similarly to the current internet, divided uniformly among the ~60000 autonomous systems, this amounts to ~2.5GB per autonomous system. Our design assumed a simple one DFZ prefix per autonomous system to achieve this uniformity, but a more sophisticated advertising scheme may break this uniformity. For example, placing popular prefixes in the DFZ would lead to ASs with popular data storing a greater proportion of the GNRS data.

We also consider the network traffic overhead of MNDN. In addition to the assumption above, we also assume that each mobile device moves approximately 100 times per day. This means that the 150TB working data set of the GNRS is updated 100 times a day which leads to a traffic overhead is ~14Gb/s. This is miniscule compared to the ~30000Gb/s of *mobile* traffic as of 2016 [1].

7 CONCLUSION

In this paper, we proposed MNDN to address the joint concerns of scalability and mobility. This is achieved by separating the slowly changing information about whether a name prefix has the potential to be mobile from information about its current location, noting the difference in dynamism of those two types of data. We achieve this by composing the DMap service from MobilityFirst with SNAMP from NDN, concluding that the divergent objectives of scalability and mobility can be jointly supported in this design. We point the reader to several important directions for future work: more extensive evaluation of performance; security, and especially denial of service; and an extended set of metrics for evaluating the conjunction of concerns of scalability and mobility.

ACKNOWLEDGEMENT

This work was supported by the NSF under Grant No. 1413973.

REFERENCES

[1] 2017. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2016-2021.* Technical Report 1454457600805266.

[2] A. Afanasyev, X. Jiang, Y. Yu, J. Tan, Y. Xia, A. Mankin, and L. Zhang. 2017. NDNS: A DNS-Like Name Service for NDN. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. 1–9. https://doi.org/10.1109/ICCCN.2017.8038461

[3] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and L. Zhang. 2013. *Scaling NDN Routing: Old tale, new design.* Technical Report NDN-TR-0004. UCLA, Los Angeles, CA, USA.

[4] Alexander Afanasyev, Cheng Yi, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2015. SNAMP: Secure namespace mapping to scale NDN forwarding. In *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*. IEEE, 281–286.

[5] RJ Atkinson and RJ Bhatti. 2012. *Identifier-Locator Network Protocol (ILNP) Architectural Description.* Technical Report RFC 6740.

[6] J. AugÃĬ, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, and X. Zeng. 2018. MAP-Me: Managing Anchor-Less Producer Mobility in Content-Centric Networks. *IEEE Transactions on Network and Service Management* 15, 2 (June 2018), 596–610. https://doi.org/10.1109/TNSM.2018.2796720

[7] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang. 2014. VANET via Named Data Networking. In *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 410–415. https://doi.org/10.1109/INFCOMW.2014.6849267

[8] Y. Hu, R. D. Yates, and D. Raychaudhuri. 2015. *A Hierarchically Aggregated In-Network Global Name Resolution Service for the Mobile Internet.* Technical Report WINLAB-TR-442. Rugtgers - The State Univeristy of New Jersey, North Brunswick NJ, USE.

[9] V. Jacobson, R. L. Braynard, T. Diebert, P. Mahadevan, M. Mosko, N. H. Briggs, S. Barber, M. F. Plass, I. Solis, E. Uzun, B. Lee, M. Jang, D. Byun, D. K. Smetters, and J. D. Thornton. 2012. Custodian-based information sharing. *IEEE Communications Magazine* 50, 7 (July 2012), 38–43. https://doi.org/10.1109/MCOM.2012.6231277

[10] Van Jacobson, D. K. Smetters, James D. Thornton, Michael Plass, Nick Briggs, and Rebecca L. Braynard. 2009. Networking named content. In *In CoNEXT âĂŹ09: Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 1–12.

[11] Do-hyung Kim, Jong-hwan Kim, Yu-sung Kim, Hyun-soo Yoon, and Ikjun Yeom. 2012. Mobility Support in Content Centric Networks. In *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking (ICN '12)*. ACM, New York, NY, USA, 13–18. https://doi.org/10.1145/2342488.2342492

[12] X. Liu, W. Trappe, and J. Lindqvist. 2014. A Policy-driven Approach to Access Control in Future Internet Name Resolution Services. In *Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture*. ACM, 7–12.

[13] X. Liu, W. Trappe, and Y. Zhang. 2013. Secure Name Resolution for Identifier-to-Locator Mappings in the Global Internet. In *Proceedings of the 22nd International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–7.

[14] P. Mockapetris. 1987. *Domain names - concepts and facilities.* STD 13. RFC Editor. http://www.rfc-editor.org/rfc/rfc1034.txt

[15] P. Mockapetris. 1987. *Domain names - implementation and specification.* STD 13. RFC Editor. http://www.rfc-editor.org/rfc/rfc1035.txt

[16] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson. 2008. *Host Identity Protocol.* Technical Report RFC 5201.

[17] Xavier Mwangi. 2018. *Scalable Mobility in Future Internet Architectures.* Master's thesis. Massachusetts Institute of Technology, Cambridge, MA. USA.

[18] Dipankar Raychaudhuri, Kiran Nagaraja, and Arun Venkataramani. 2012. MobilityFirst: A Robust and Trustworthy Mobility-centric Architecture for the Future Internet. *SIGMOBILE Mob. Comput. Commun. Rev.* 16, 3 (Dec. 2012), 2–13. https://doi.org/10.1145/2412096.2412098

[19] R. I. Rivest and B. Lampson. 1996. SDSI - a simple distributed security infrastructure. In *Proceedings of the 18th Annual International Cryptology Conference*.

[20] Abhigyan Sharma, Xiaozheng Tie, Hardeep Uppal, Arun Venkataramani, David Westbrook, and Aditya Yadav. 2014. A Global Name Service for a Highly Mobile Internetwork. In *Proceedings of the 2014 ACM Conference on SIGCOMM (SIGCOMM '14)*. ACM, New York, NY, USA, 247–258. https://doi.org/10.1145/2619239.2626331

[21] Reza Tourani, Satyajayant Misra, Travis Mick, and Gaurav Panwar. 2018. Security, Privacy, and Access Control in Information-Centric Networking: A Survey. *IEEE Communications Surveys and Tutorials* 20, 1 (2018), 66 – 600. https://doi.org/10.1109/COMST.2017.2749508

[22] Arun Venkataramani, James F. Kurose, Dipankar Raychaudhuri, Kiran Nagaraja, Morley Mao, and Suman Banerjee. 2014. MobilityFirst: A Mobility-centric and Trustworthy Internet Architecture. *SIGCOMM Comput. Commun. Rev.* 44, 3 (July 2014), 74–80. https://doi.org/10.1145/2656877.2656888

[23] T. Vu, A. Baid, Y. Zhang, T. D. Nguyen, J. Fukuyama, R. P. Martin, and D. Raychaudhuri. 2012. DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet. In *IEEE ICDCS*. IEEE.

[24] Liang Wang, O. Waltari, and J. Kangasharju. 2013. MobiCCN: Mobility support with greedy routing in Content-Centric Networks. In *2013 IEEE Global Communications Conference (GLOBECOM)*. 2069–2075. https://doi.org/10.1109/GLOCOM.2013.6831380

[25] S. Yang, J. Kurose, S. Heimlicher, and A. Venkataramani. 2015. Measurement and modeling of user transitioning among networks. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 828–836. https://doi.org/10.1109/INFOCOM.2015.7218453

[26] Yu Zhang, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. 2016. A survey of mobility support in named data networking. In *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*. IEEE, 83–88.