ICN Research Group
Internet-Draft
Intended status: Informational
Expires: January 3, 2019

A. Azgin
R. Ravindran
Huawei Technologies

July 2, 2018

DMS: Dynamic Inter- and Intra-Domain Mobility Support Framework for Information Centric Networking
draft-azgin-icnrg-mobility-00

Abstract

The objective of this proposal is to introduce a mobility support framework for information centric networking (ICN) that is capable of supporting dynamic mobility scenarios associated with both consumer and producer applications, in a mobile device connected to a wireless LAN or WAN point of attachment (PoA). Due to rapidly evolving user trends that shift towards increased mobility and increased access to mobile content delivery (as both content producer and consumer), it is essential for an ICN architecture to offer active mobility support to end hosts, and present them with varying degrees of quality of experience. We enable this through the use of a network-driven mobility support architecture, which operates in a distributed and anchorless manner, and relies on late-binding and in--band signaling with the use of forwarding labels.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 3, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. Related Work 4
  2.1. Consumer Mobility 4
  2.2. Producer Mobility 4
3. Mobility Requirements 6
4. Design Components 7
  4.1. Mobility Service and Components 7
  4.2. Forwarding Label and Components 8
  4.3. Distributed Mobility Controller and Components 9
5. DMS Protocol Design 9
6. DMS Implementation 11
  6.1. Registration Phase 11
    6.1.1. Producer Mobility 11
    6.1.2. Consumer Mobility 15
  6.2. Content Delivery Phase 16
  6.3. Handover Phase 18
    6.3.1. Intra-domain Handover 18
      6.3.1.1. Producer Mobility 18
      6.3.1.2. Consumer Mobility 20
    6.3.2. Inter-domain Handover 21
7. Design Discussions 22
  7.1. Storage Considerations 22
  7.2. Scalability Considerations 23
  7.3. Security Considerations 23
    7.3.1. Producer Flooding 24
    7.3.2. Consumer Flooding 24
8. Informative References 24
Appendix A. Additional Stuff 26
Authors' Addresses 26

1. Introduction

Mobility support for information centric networking (ICN), with specific emphasis on content-centric networking (CCN) [CCN], continues to be a critical research area [MOB12] [MOB16] to ensure its viability in diverse and dynamic networking environments, while

offering an acceptable quality of service (QoS) to mobile users as good as or better than the current solutions. Default mobility support in CCN is a best-effort service that relies on its pull-based design using request-data primitives, with end hosts receiving content objects after making a request for them by sending Interest messages towards the source. In this design, consumer mobility is handled reactively, as the consumer re-expresses interests by making retransmission requests for the dropped (or undelivered) content objects due to the host's mobility, with in-network caching helping to speed up recovery, however, with no guarantees on service quality such as packet loss or content delivery latency.

Support for producer mobility, on the other hand, is not included by design, as the default approach to re-discover producer's location after a handover would trigger a timeout-based broadcast storm, which would be unrealistic in practical scenarios. For this reason, multiple approaches have been proposed to offer such support [KITE][FWLRP][ANCLS] and provide some guarantees for end user experience, as the lack of it can create significant issues [NDNMS]. These solutions have focused on handling producer mobility using various designs: (i) an anchor based solution [KITE] with requests being forwarded through network or application specific anchor points, (ii) a network-driven locator-based solution that separates persistent application and topological network identifier namespaces while relying on the late-binding technique [FWLRP], and (iii) an anchorless host-driven design [ANCLS] with the mobile node responsible for notifying the network of its movements to trigger temporary updates on data plane. Host or user entity (UE) driven mobility mechanisms such as [KITE] and [ANCLS] offer poor support to achieve a make-before-break solution, as the path correction in these mechanisms is only possible after the UE attaches to the new PoA, while network driven solution can proactively start forwarding the Interests from the UE's old location to the new one immediately after it learns about the UE's detachment process.

There are advantages and disadvantages to each of these earlier solutions, however, to address the specific application needs by offering performance guarantees to them in a scalable manner, we believe a network-driven solution is needed [MAAS]. Furthermore, to provide a complete mobility solution for ICN with similar performance guarantees, we cannot rely on the default CCN architecture, as the level of mobility support offered by CCN is minimal, at best. Therefore, we need solutions that can offer performance guarantees in regards to latency and throughput, regardless of the type of application a host runs, be it consumer or producer. The architecture we propose here achieves this objective by using a comprehensive mobility framework that combines active producer mobility support with active consumer mobility support. Proposed

approach takes advantage of locator-driven forwarding with the use of forwarding label [I-D.ravi-icnrg-ccn-forwarding-label], which is an optional hop-by-hop header carried within the Interest messages to identify a topological name (i.e., a network domain, or a network host). As a result, proposed approach can offer seamless mobility support to end hosts.

2. Related Work

2.1. Consumer Mobility

Existing research on consumer mobility support for CCN/NDN is limited due to its limited impact on the performance, as the default best-effort approach considered for CCN/NDN can offer some limited guarantees. Available research on consumer mobility can typically be categorized as proactive solution or reactive solution. Proactive approach can target upcoming mobility events to for instance cache content at nearby locations for faster access to content, before the handover takes place. On the other hand, reactive approach is triggered after a handover takes place. Default CCN/NDN approach falls in the latter category.

For instance, [MOBINDN] uses a centralized architecture to support host mobility, which, during/after handover, updates forwarding table entries corresponding to request namespaces to create a shorter retransmission path for the consumer requests from the new point of attachment to the old one. The approach considered in [CMOB16] uses prediction on consumer location to detect handover events and estimate its location at future time points. The approach also predicts which Interest packets will be dropped during a handover, and proactively cache Data packets associated with them at nearby routers so as to reduce retrieval latency after a successful handover.

Even though the existing approaches can offer faster access to Data packets that can improve the perceived quality of experience, no guarantees can be offered to ensure certain quality requirements are met, as network support is limited.

2.2. Producer Mobility

To improve the performance associated with Producer mobility in NDN/CCN-based ICN architectures, various approaches have been proposed, and these approaches can be grouped into three categories:

- First, we have anchor-based solutions [KITE], where requests are forwarded through designated nodes (i.e., network or application specific nodes) that keep a record of mobile node movements or

provide a path towards it. As path setup requires setting up and maintaining traces within the network, such an approach can introduce significant signaling and state overhead. Furthermore, gradual updates may prevent support for seamless mobility and the use of application-specific anchors can regularly lead to excessive path stretches.

- Second, we have anchor-less mobility solutions [ANCLS], where the mobile node is responsible for notifying the network of its movements by, for instance, triggering temporary updates on data plane. In [ANCLS], temporary FIB entries are created at the routers to override the default forwarding strategy to forward packets towards the mobile Producer. However, as the approach relies on existing schemes to provide route updates and name resolution, local updates triggered by mobile handovers continually result in local routing churn, while the global routing convergence creates more instability in the forwarding paths. As the number of mobile hosts increases with increased traffic rate, scalability and global routing convergence can be a major concern. Furthermore, such an approach would require full support from the network to temporarily override the existing FIB entries, which may only be possible when appropriate security mechanisms to handle such signaling are applied. In such case, seamless mobility cannot be guaranteed.

- Third, we have late-binding approaches [MAAS], which relies on splitting the application and network identifier namespaces, and where dedicated nodes that keep track of mobile movements are used to resolve content or device prefixes or identifiers to locators, which are then used as forwarding label to forward the requests. For instance, [MAAS] utilizes dedicated controllers in combination with in-band signaling that takes advantage of Interest-Data symmetric routing within each domain to handle name resolution management. The local controllers can communicate with other controllers (such as home controllers) to handle inter-domain mobility. While this approach mitigates path stretch issue, it does increase the signaling overhead due to mobile end points. Furthermore, controllers can become a bottleneck as the failure of a local controller would lead to either temporary disconnects or increased overheads. Lastly, even though [MAAS] can support seamless Producer mobility with minimal path stretch, it requires architectural changes that can go beyond what might be considered for [KITE] and [ANCLS] as incremental upgrades to support seamless mobility.

3. Mobility Requirements

Locator identifier split can be considered as an important requirement to handle mobility at the network layer [SEAML], which can inherently be supported by the ICN architectures (such as MobilityFirst [MFRST]) due to unique names assigned to each entity and routing on names to resolve locations using a global name resolution system along with delay tolerant networking (DTN) features such as hop-by-hop forward and store, and late-binding, to achieve seamless connectivity. There are some limitations to achieve these objectives efficiently with the current IP through the use of protocols based on Mobile IP, due to using IP address both as a locator and as an identifier, triangular routing and the associated control overhead [RFC6301].

In CCN/NDN based architectures, using hierarchical identifiers for routing cannot scale due to potentially explosive growth in namespaces and the reduced aggregatability of these namespaces with increased mobility, multi-homing, or resource replication [NCMP]. Furthermore, practical problems such as name-suffix hole may arise when names are used for network reachability. Routing scalability is typically achieved by designing names with aggregatable property, which is the case for IP today. However, having such feature in CCN/NDN would lead to relinquishing the persistency of names, as the names would involve a topological component for scalability (which also suggests resources to be renamed depending on, for instance, network or business specs or characteristics). Furthermore, overloading an identifier as a locator can lead to unstable routing control and forwarding plane operations.

Locator identifier split in CCN/NDN therefore imposes splitting the hierarchical namespace to support routable, persistent and human-friendly names. In such case, names would be divided based on application binding vs. advertised network entities in control plane to improve the scalability of routing. For instance, persistent identifiers, which would be used to create secure content objects, can be published by multiple content distributers, where they would then be mapped to different locators to resolve the content names to specific infrastructure entities. The fundamental requirement with this form of splitting is no different than that of MobilityFirst or LISP [RFC6830], which is the requirement of a name resolution system (NRS) to map the two namespaces.

In addition to the locator identifier split, we can summarize the general requirements for CCN/NDN based architectures to support efficient packet forwarding involving mobile hosts as follows:

- o Forwarding scalability: The size for the forwarding tables should be independent of the number of mobile entities and the level of mobility experienced by these entities. This requirement can be met by requiring to keep object state only at the edges (i.e., gateways), and perform routing at the core network based on the locator prefixes.

- o Control overhead: Network updates triggered by mobility should be local rather than global, and they should not affect the convergence of routing/forwarding. This requirement can be met by using a localized mobility controllers (within each domain/AS) that is capable of resolving namespaces corresponding to mobile entities using the corresponding home bindings.

- o Intra- and Inter-session mobility: Mobility support architecture should be capable of supporting both intra-session mobility (e.g., changing access point or mobility service node during an active communication instance) and inter-session mobility (e.g., reconnecting after timeout or session expiry). We can support inter-session mobility through reactive discovery by communicating with a mobility controller associated with the mobile entity's namespace (i.e., home mobility controller, to which the mobile is registered globally for its namespace and which provides global resolution on such registered namespaces) to request an update. We can support intra-session mobility through active tracking of the mobile entity's movements, for instance, using proactive signaling by the mobile entity to trigger update at all the related ICN forwarders along the reverse Data path signaling to re-resolve the namespace for the mobile entity to the correct locator.

- o Mobility granularity: Since supporting mobility may incur increased signaling and communication overhead, it should be realizable as a service, on-demand and for a subset of the mobile entities (e.g., hosts/namespaces subscribed to the mobility service). Granularity for the mobility service can be further adjusted with respect to the geographical span of mobility support, and latency, etc.

4. Design Components

4.1. Mobility Service and Components

While the proposed solution can be applied to all Interest flows in a mobile network, it can also be selectively enabled to only mobile producers. For this, our architecture relies on the use of mobility service for matching entities (such as hosts, service, or content name prefixes). The presence of mobility service is identified using

one of the following approaches: (i) with the use of Mobility Service flag carried within the Interest which can be set by the end hosts or ICN-enabled point of attachments (default approach for our architecture), (ii) by prepending the content name prefix with the Mobility Service tag, or (iii) by provisioning traffic policy rules at the ICN service routers to monitor for Interest flows with pre-configured names matching a prefix of the Interest (or by checking the Interest's attributes to distinguish flows requiring seamless mobility support). If mobility service is enabled on a namespace, DMS protocols are used on the received Interests or forwarded Data packets to provide seamless mobility support for the matching flows. If mobility service is not enabled, regular CCN/NDN processing logic is used on the received Interests or Data packets, by going through the default CS/PIT/FIB processing.

4.2. Forwarding Label and Components

To realize ID/locator split in CCN, we use the forwarding label (FL) object that acts as a locator and provides the flexibility to forward Interests on a name other than the one provided within the original Interest, while allowing the ability to modify it on-the-fly [I-D.ravi-icnrg-ccn-forwarding-label]. Proposed FL-object helps with not only mobility but also with opportunistic routing, binding Interests to services at a given location, and in-network computing, thereby allowing for incremental enhancement over CCN to provide richer and optimized service aware routing at the network edge.

FL objects are considered as container objects that include locator identifier, service specific metadata (i.e., contextual information on the application/service to help the network triggering appropriate FL processing such as trust validation) and (optional) security attributes for authentication. Locator identifier is considered as a hierarchically structured topological name representing a domain, a gateway, or host identifiers.

FL objects are inserted within the Interest either by the consuming application (which may require a trust binding between the content identifier or prefix and the locator identifier) or by the network (which typically occur at the ingress service routers, if the Interest satisfies an existing flow service profile). As an FL object can be modified within the network (e.g., at domain boundaries, network edges, etc.), it is considered as part of the optional hop-by-hop header. In regards to processing of FL-object carrying Interests, various options are available depending on service profiles and trust relationships, e.g., locator identifier preference (over content identifier), content identifier preference (over locator identifier), locator identifier preference with swap, or locator identifier ignore.

For efficient utility and processing of FL objects, a separate data structure is introduced at the ICN nodes, referred as the Forwarding Label Table (FLT), which carries locator/identifier cache mappings for supported flows. Packet processing logic for the FLT corresponding to mobile flows is similar to that of FIB, which suggests longest prefix matching on the carried content identifiers to extract the locator information, which is then used for exact match on the FIB. Also note that, utilization of FL objects is not limited to mobile flows, as they can also be used to provide fast forwarding path such as off-path caching on supported flows.

4.3. Distributed Mobility Controller and Components

Distributed Mobility Controllers (DMCs) are localized service nodes within domains to handle mobility support for subscribed or visiting mobile entities. DMC nodes operate in a decentralized manner to resolve content/host identifiers to locators. These nodes use a Local Registration Database (LDB) to store name to locator mappings retrieved through local registrations (e.g., for visiting mobile entities) and/or remote registrations (e.g., for member mobile entities). DMC nodes also communicate with other DMC nodes in different domains, using Route Request (RREQ) and Route Response (RRES) messages, to discover current locator information on mobile entities. DMC nodes are also responsible for updating the service points and gateway nodes within their domain for localized name-to-locator mappings.

5. DMS Protocol Design

Leveraging ICN's name-based mobility, the proposed architecture allows any meaningful space of the name hierarchy to be mobile. End hosts enable this by actively registering the associated namespace to the network, where the mobility service enabled by the provider allows the entities in another domain under the namespace to be accessible anywhere on the Internet. This motivates a scalable mobility architecture. The proposed solution achieves this objective by utilizing four essential building blocks: Distributed Mobility Controllers (DMCs) to provide name-to-locator mappings and manage control overhead, Forwarding Label Table (FLT)s, which are used by the designated routers to control information flow in the network) to address forwarding scalability, Forwarding Labels and Mobility Tags to address inter- and intra-session mobility at different mobility granularities.

Proposed architecture considers a basic networking hierarchy, which consists of a given number of domains or Autonomous Systems (AS). Each domain/AS is assigned a DMC carrying a domain designated prefix. DMCs are responsible for mapping local and remote entity names to

domain/router identifiers. If the entity is local (which is the case for intra-domain delivery), DMC resolves the names to ISRs, whereas, if the entity is remote (which is the case for inter-domain delivery), DMC maps the names to local domain's egress router identifier. Each host is statically or dynamically assigned to a designated DMC (also referred as Home-DMC), which stores up-to-date resolution information regarding its users.

Assuming the use of hierarchical identifiers, each host is assigned a unique identifier representing the association of a user to its home network. The aggregate identifier (including network and host identifiers) is only needed during Registration. Network identifier, for an endpoint, is not considered a priori requirement to support successful location discovery. Using an identifier, however, is the preferred choice to minimize control overhead.

DMCs are assumed to have access to information directly related to communication taking place within its domain (e.g., Home DMC or locator associated with content published within the domain by current registered hosts). Hence, no direct synchronization is assumed to exist among the DMCs. Active domain information on visiting hosts is flushed on a regular basis, as soon as the registered host leave the domain under DMC's control, to minimize access to outdated information. However, Remote-DMCs are allowed to store information on Home-DMCs representing the visiting hosts for longer periods to minimize overhead associated with the initial discovery (i.e., an ISR requesting a mapping for a consumer residing in Remote-DMC's domain on a previously hosted Producer's prefix, or during decentralized discovery responding with information on Home-DMC to a request from another domain's DMC).

In this architecture, forwarding labels are utilized to route packets in a controlled manner. Forwarding label is similar to content prefix in the sense that, when used, it provides sufficient information on the next physical or logical hop. However, unlike a content prefix, forwarding label is used as a dynamic tag within the Interest that is regularly updated (at the service routers and gateway points) along the path to content source. To support the use of forwarding labels, at each service or edge router or PoA, we utilize an FLT that carries the mappings corresponding to content prefixes and forwarding addresses.

Since handling mobility with FLT and forwarding labels incurs additional memory and computational costs, forwarding decisions based on FLT are limited to traffic tagged as being part of the mobility service, whereas for other traffic, default routing based on FIB is used. Therefore, any mobile host that wishes to use the mobility service indicates its intent during the registration by setting a

single-bit mobility service tag (MS-tag) contained within the registration message. Consumers can also enable this tag along with the mobile entity's unique name to invoke the mobility service, which would help ISRs to differentiate between mobile and non-mobile entities. In scenarios where the mobility service is not explicitly defined, forwarding is strictly based on the core CCN/NDN policies. Note that, the use of mobility service also allows the use of in-network caching and look-up based on the available FIB entries.

6. DMS Implementation

In this section, we explain the general operations for the proposed DMS architecture in regards to (i) how the namespaces are registered for faster resolution and delivery, (ii) how the content is delivered from/to mobile end hosts, and (iii) the operations involved during a handover.

6.1. Registration Phase

6.1.1. Producer Mobility

```
                            +--------------+
                            |              |
                            | +----------+ |
       +------------------> | | HomeDMC  | | <-------------------+
       |                    | +----------+ |                     |
       | Resolution         |              |         Update      |
       |                    +--------------+                     |
       |                         HomeAS                          |
       |                       (Producer)                        |
       |                                                         |
   +----------------------------+        +-----------------------------+
   | +-----+                    |        |                         |   |
   | | DMC |             +------+        +------+             +---+--+ |
   | +--+--+      +----> |EISR |-------> ||IISR |      +----> |  DMC | |
   |    ^         |      +------+        +---+--+      |      +------+ |
   |    |         |             |        |   |         |               |
   |    |         |             |        |   |         |               |
   |    |      +--+---+         |        |   |         |               |
   |    +------+ ISR2 |         |        |   |     +---+--+            |
   |           +---+--+         |        |   +---> | ISR1 |            |
   |               |            |        |         +--+---+            |
   |               |            |        |            |                |
   |           +---+--+         |        |         +--+--+            |
   |           | PoA2 |         |        |         | PoA1|             |
   |           +---+--+         |        |         +--+--+            |
   |               |            |        |            |                |
   |               |            |        |            |                |
   |         +-----+----+       |        |       +----+-----+          |
   |         | Consumer |       |        |       | Producer |          |
   |         +----------+       |        |       +----------+          |
   +----------------------------+        +-----------------------------+
             ConsumerAS                                  CurrentAS
```

Figure 1: Basic view for the mobility support architecture

We start by assuming that the Producer publishes content under namespace /Prefix, and the end hosts learn the minimum control name space needed to interact with the access points (or PoAs) during mobile attachment. Figure 1 illustrates the basic topology. Here, ISRs can be considered as the service points for the end hosts connected to PoA(s) under each.

Step(1): Registration Phase initiates with Producer sending a Register (REG) message for its content, under namespace /Prefix, to the current host domain's DMC (e.g., CurrentDMC). REG message includes the complete identifier for the Producer's namespace, including the bindings for it: /Prefix:/HomeAS/ProducerID, where

/HomeAS/ProducerID represents the home binding identifier for the entity /Prefix, with /HomeAS representing the identifier for Producer's home domain. Here, for instance, /ProducerID can be the device identifier, and if the device itself is mobile then /ProducerID can be considered as part of /Prefix, in which case /HomeAS would be sufficient for the home binding. We place no restriction on /Prefix as for it to be globally routable or not, as it is resolved using the decentralized DMC-driven infrastructure. To complete registration, Producer sends a REG message to its PoA (e.g., PoA1) with the prefix /PoA1/REG used as a forwarding label, which is then forwarded using the existing forwarding table (FIB/FLT) entries towards CurrentDMC through PoA1 and its service point (e.g., ISR1). Note that, Producer can also send the registration message directly to DMC by using /DMC/REG (in which case, on-path nodes would decide accordingly to register). After receiving the REG message, ISR1 also updates the forwarding label by including its identifier within packet header to inform CurrentDMC on the identity of the service point (or ISR) associated with the Producer's to-be-registered namespace. After CurrentDMC receives the REG message, if the MS-tag is set within the request to indicate a request for mobility support, CurrentDMC's local registration database (LDB) is updated with the entry /Prefix::/ISR1::/HomeAS (through insertion of a new entry or modification of an existing entry), where the third component specifies the home domain for /Prefix. Here, FLT in the PoA or LDB in the DMC by default require at least two inputs, corresponding to the prefix and locator information. For the sake of presentation here, we use double colon to separate these entries. Additionally, if the MS-tag is set within the forwarded REG message, FLT tables at the on-path POA1 and ISR1 are also updated by inserting the corresponding mappings targeting /ProducerID at the PoA1, and /PoA1 at ISR1. Specifically, this is done by adding the entry /Prefix::/Producer to FLT table at PoA1, and the entry /Prefix::/PoA1 to FLT table at ISR1, upon receiving the acknowledgement message from CurrentDMC.

```
+--------------------------+            +--------------+
|                          |            |              |
|                 +------+ |    HREG    | +----------+ |
|         +-------+  DMC | | +--------> | | HomeDMC  | |
|         |       +------+ |            | +----------+ |
|         |                |            |              |
|         |                |            +--------------+
|         |                |                 HomeAS
|      +--+--+             |                    +
|      | ISR1|             |                    |
|      +--+--+     ^       |               FREG |
|         |        |       |                    ^
|      +--+--+     | REG   |           +--------+-------+
|      | PoA1|     |       |           |                |
|      +--+--+     |       |           | +------------+ |
|         |        |       |           | | PreviousDMC| |
|         |        +       |           | +------------+ |
|    +----+-----+          |           |                |
|    | Producer |          |           |                |
|    +----------+          |           |                |
+--------------------------+           +----------------+
        CurrentAS                         PreviousAS
```

Figure 2: Registration for Producer mobility

Step(2): Proactive Update Phase initiates with CurrentDMC sending a Route Update (RUPD) message to the ingress routers within CurrentAS to update their FLT tables with the entry /Prefix::/ISR1 so that any Interest received by the ingress nodes targeting the namespace /Prefix can be immediately forwarded to ISR1. At the ingress routers, in addition to proactive updates, we can also use reactive updates, which would be triggered after (i) a first request arrival, in which case the ingress router would make a request for the forwarding information after receiving an Interest under namespace /Prefix, for which no entry exists within the ingress router's forwarding tables; or (ii) receiving a mobility update on data plane, with information carried within returned Data packets. However, the reactive update phase may require the initial traffic to be forwarded to the CurrentDMC to minimize latency, which would act as a temporary anchor, hence introduce additional overhead to keep track of entries to avoid sending recurring route requests for the non-local Producers. For this reason, by default, we limit DMC to function specifically as a resolution server, rather than offering a combination of resolution and anchor functionalities.

Step(3): Home Registration Phase initiates with CurrentDMC sending to DMC within Producer's HomeAS (HomeDMC) a Home-Register (HREG) message

for /Prefix using the complete identifier information for the Producer. We can use the following as the prefix for the HREG message: /HomeAS/DMC/HREG, which assumes /DMC to be a globally shared anycast identifier within a domain. After HomeDMC receives the HREG message, it updates its LDB accordingly; (i) if an active entry is found in HomeDMC's LDB for Producer, corresponding to a different domain, the entry is updated to represent the domain change as follows: /Prefix::/Remote:CurrentDomainID::/Remote:PreviousDomainID, with the first component representing the mobile entity's namespace, the second component pointing to /CurrentAS and the third component pointing to the previous domain the Producer was visiting (e.g., /PreviousAS). Additionally, HomeDMC sends a Flush-Register (FREG) message to the PreviousDMC using the identifier /PreviousAS/DMC to timeout the existing entries while re-directing on-the-fly traffic to the current domain Producer is visiting; (ii) if no active entry is found in HomeDMC's LDB for Producer, then HomeDMC's LDB is updated with the entry /Prefix::/Remote:CurrentDomainID::/.

6.1.2. Consumer Mobility

We start by assuming that the Consumer is assigned a unique identifier /ConsumerID.

Step(1): Upon connecting to a network, Consumer sends a local registration request (LREG message) towards a matching entity as assigned by the hosting domain to register at the point of attachment (PoA2) or the service point (ISR2) using its identifier /ConsumerID (which could be a device ID, or a set of service IDs if service level mobility is requested) by requesting mobility service support from the network. The ConsumerID is also appended to the Interest, but its scope is limited only to the link between the consumer and the PoA.

Step(2): Upon receiving the LREG message, PoA2 or ISR2 registers /ConsumerID within its FLT to include information on the Consumer, such as its identifier, interface identifier, and any associated forwarding label.

```
     +-----------------------------------------+
     |                                         |
     |           +-----+       +-------------+ |
     |           | ISR2| +---> | ConsumerDMC | |
     |         ^ +-----+       +-------------+ |
     |         |     |                         |
     |         |  +-----+                      |
     |     LREG|  | PoA2|                      |
     |         |  +-----+                      |
     |         |     |                         |
     |         +     |                         |
     |          +----------+                   |
     |          | Consumer |                   |
     |          +----------+                   |
     +-----------------------------------------+
                   ConsumerAS
```

Figure 3: Registration for Consumer mobility

6.2. Content Delivery Phase

In this section, we present the default content delivery for the proposed architecture in the case of no handover.

Step(1): Consumer starts Content Delivery Phase by sending an Interest for /Prefix to its PoA (PoA2), while including its host identifier /ConsumerID within the request. We assume the request is a fresh request, with no existing entry within PIT or a matching content within CS along the path towards the Producer. If one exists, default CCN/NDN operations would take place (i.e., if PIT entry exists, aggregate by including /ConsumerID, if content exists, respond with it). We also assume PoA acts as the service point for the end host with mobility service installed.

Step(2): PoA2 checks its CS and PIT to find a matching entry for /Prefix, and finds no match. PoA2 creates an entry within its PIT for the request to also include /ConsumerID. PoA2 forwards the Interest to its service point (ISR2) after removing /ConsumerID. After receiving the Interest, ISR2 performs a more detailed check including searching for FLT (and, if necessary FIB) entries. If no entry is found, ISR2 creates a Route-Request (RREQ) message with the prefix /DMC/RREQ and sends it to its local DMC. Furthermore, ISR2 updates its local request waiting list (LRWL) for the request messages it creates (similar to the PIT entries) to avoid retransmitting the same request to its DMC. Any further Interests targeting /Prefix are queued at ISR2 until a response to the first RREQ message is received on the given namespace.

Step(3): After DMC receives the RREQ message, it searches for a matching entry within its LDB. If an entry is found, request can be unicast sent to the HomeDMC for /Prefix. Similarly, request message can be forwarded to HomeDMC, if /Prefix includes a domain-specific identifier. If no entry is found, and the received content name does not identify the home binding, then the request would use the decentralized DMC infrastructure to identify HomeDMC for /Prefix. One approach would be to forward the RREQ message to all the other DMCs. We can reduce the discovery overhead by using a controlled name-based multicast, by grouping multiple requests into a single Interest and forwarding the Interest domain by domain until a match is found. However, in practice, we can expect such home mapping to be provided at the time of request. Also, similar to how it was with ISR2, DMC also implements a LRWL for the received RREQ messages, to suppress discovery attempts associated with new requests targeting the same namespace.

Step(4): After HomeDMC receives the RREQ message, a Route-Response (RRES) message is created in response to the request with the mapping /Prefix::/CurrentAS that identifies the current location for the Producer. Note that, if the response message is forwarded along trusted domains, such responses can also be cached within on-path domains to improve discovery timeframe for future requests.

Step(5): After Consumer side DMC receives HomeDMC's RRES message to its RREQ message, DMC first updates its LDB with the entry /Prefix::/CurrentAS::/HomeAS, identifying the current location for the Producer and its home domain. DMC also creates a Data packet in response to the received RREQ messages and alerts any ISR nodes indicated by the LRWL entry associated with /Prefix, before removing any such entries. For the given scenario, DMC responds to ISR2's RREQ message by creating a RRES message with the mapping /Prefix::/CurrentAS::/ER1, by also including information on the preferred egress point (ER1), to which the ISR2 needs to forward the Interest messages. Note that, by default, such mapping information may readily be available at the ISR nodes within their FIB.

Step(6): After ISR2 receives the RRES message corresponding to its earlier request, ISR2 also updates its FLT with the following entry: /Prefix::/CurrentAS, while updating its FIB with the entry /CurrentAS::/ER1. ISR2 then clears its LRWL entry for /Prefix and starts forwarding the matching Interests using the forwarding label /ER1/CurrentAS (with first destination being /ER1 and final destination residing in /CurrentAS).

Step(7): After ER1 receives an Interest carrying a forwarding label, it extracts the forwarding label to determine the next hop. In the current scenario, ER1 determines itself as the first destination,

checks for the next destination and identifies it as /CurrentAS. Assuming that the ERs have already populated their FIBs with domain-based mappings of /RemoteAS::/NextHop, ER1 can set the received Interest's forwarding label to /NextHop/CurrentAS before forwarding it to /NextHop. This process is repeated until the Interest is received by the ingress router at CurrentAS.

Step(8): After the ingress router at CurrentAS receives the Interest, it determines that the target for the received Interest resides within its domain. Ingress router then uses its FLT to determine the address for the ISR servicing the PoA connected to the Producer hosting /Prefix. FLT-based lookup process is repeated at the service point and point of attachment, until the Producer receives the Interest, after which content delivery towards the Consumer can proceed along the reverse path, using CCN/NDN's breadcrumb approach.

Step(9): After the content object arrives at the PoA2, it searches for and extracts to PIT entry to determine (i) the consumer identifier, if it exists, and (ii) the interface metrics. For the case that the consumer identifier (/ConsumerID) exists, PoA2 performs a lookup on the FLT to find the entry for /ConsumerID to validate the interface information. For the default case, with no handover, interface information within FLT entry would be equal to the interface metric extracted from the PIT entry, thereby allowing the content object to be forwarded using the default forwarding process.

6.3. Handover Phase

6.3.1. Intra-domain Handover

6.3.1.1. Producer Mobility

Producer performs a handover, and connects to PoA3.

```
      +-------------------------------------------+
      |                                           |
      |              +------+                     |
      |      +-------+  DMC +-------+             |
      |      |       +------+       |             |
      |      |                      |             |
      |      |                      |             |
      |      |                      |             |
      |   +--+--+               +---+--+          |
      |   | ISR1|               | ISR3 |   ^      |
      |   +--+--+               +---+--+   |      |
      |      |                      |      |      |
      |   +--+--+               +---+--+   |      |
      |   | PoA1|   Handover    | PoA3 |   | REG  |
      |   +-----+  +--------->  +---+--+   |      |
      |                              |      |      |
      |            +----------+     |      |      |
      |            | Producer +-----+      +      |
      |            +----------+                   |
      |                                           |
      +-------------------------------------------+
                   CurrentAS
```

Figure 4: Producer handover

Step(1): After the Producer performs a handover from PoA1 to PoA3 serviced by different ISRs, Producer repeats the registration process by sending a REG message to CurrentDMC, as explained earlier. After the CurrentDMC receives the REG message, it looks up for a matching entry in its LDB for /Prefix to determine an intra-domain handover, from ISR1 to ISR3. CurrentDMC updates its LDB entry with /Prefix::/ISR3::/HomeDMC and initiates a proactive mobility update by sending a Route-Update (RUPD) message to the ingress points so that these border routers update their FLT tables with the entry /Prefix::/ISR3. CurrentDMC also creates an FREG message and sends it to ISR1, which include information on the new service point for /Prefix.

Step(2): After ingress routers receive the RUPD message, they update their FLT tables to point to ISR3 for /Prefix and start forwarding any matching Interest towards the new ISR associated with /Prefix, ISR3.

Step(3): After ISR1 receives the FREG message for /Prefix, ISR1 updates its local FLT with the entry /Prefix::/ISR3 and forwards any incoming Interest targeting this namespace to ISR3. Additionally, ISR1 also starts a timer to flush out its local FLT entry corresponding to /Prefix when the timer expires. ISR1 also forwards

the FREG message to PoA1, which flushes its FLT entry upon receiving. Note that, even though PoA1 can timeout its FLT entry after a handover notification from Producer, FREG allows confirmation from CurrentDMC of a successful registration after handover.

6.3.1.2. Consumer Mobility

```
 +-----------------------------------------+
 |            +-------------+               |
 |          +-> ConsumerDMC |               |
 |          | +------^------+               |
 |          |        |                      |
 |          |        |                      |
 |       +--+--+     +---+------+           |
 |       | ISR2|         | ISR4 |           |
 |       +--+--+         +---+--+  ^        |
 |          |                |     |        |
 |       +--+--+         +---+--+  |        |
 |       | PoA2|         | PoA4 |  |        |
 |       +-----+         +------+  |        |
 |                         +--^    |        |
 |                         |       +        |
 |            +----------+ |     LREG       |
 |            | Consumer | +                |
 |            +----------+                  |
 |                                          |
 +-----------------------------------------+
                ConsumerAS
```

Figure 5: Consumer handover

Consumer mobility requires a more proactive approach as the mobile host is the one triggering the content request.

Step(1): As Consumer initiates a handover from PoA2 to PoA4, during the handover signaling, PoA2 is informed of the next PoA (or the set of candidate PoAs) for the Consumer.

Step(2): After being informed of the upcoming handover to PoA4, PoA2 updates the existing FLT entry for the Consumer corresponding to /ConsumerID by setting the interface metric to a predefined value corresponding to Not_Attached and initializing the forwarding label entry to identifier corresponding to PoA4 (/PoA4). If the next PoA information received from Consumer includes a list of candidate PoAs, then the FLT entry would include multiple forwarding labels corresponding to these PoAs (or a smaller subset of them).

Step(3): After the Consumer connects to PoA4, if the Consumer requires mobility support from the network, it sends an LREG message to PoA4 to register its host identifier (/ConsumerID) at the PoA within its FLT.

Step(4): After Consumer's handover to PoA4, as content objects arrive at PoA2, due to Consumer's earlier Interests sent before the handover, PoA2 extracts the matching entry from its PIT and determines the use of mobility support service by a requesting Consumer, which is identified through its host identifier /ConsumerID. PoA2 then uses /ConsumerID to extract the FLT entry and determines the forwarding label associated with the requesting host. PoA2 then creates a Push-Data (PSHD) type content object, which is a special type of data packet that bypasses PIT lookups during packet forwarding. PoA2 then performs FIB lookup on the extracted forwarding label (/PoA4), inserts the label within the packet and changes its type to PSHD before forwarding it to the next hop towards PoA4.

Step(5): Intermediate nodes that receive the PSHD-type content objects, skips CS/PIT processing, and performs FIB lookup on its forwarding label to determine the outgoing interface before forwarding it to the next hop as indicated by the FIB entry. The set of nodes that do not have a corresponding PIT entry can also save this Data packet in their content stores to benefit other consumers.

Step(6): As the PoA4 receives these PSHD-type content objects, they are queued at its CS for later retrieval by the Consumer, if the process relies on the Consumer to re-express Interests for these content objects. If the received PSHD-type content objects include host identifier, PoA4 can push these content objects to the Consumer as soon as the Consumer finishes handover and successfully registers at PoA4. To do that, PoA4 performs a lookup on the FLT to validate the registration for the /ConsumerID, before forwarding towards the interface as indicated by the corresponding FLT entry.

6.3.2. Inter-domain Handover

Step(1): After Producer moves to a new domain, NextAS, the Producer initiates the registration phase by sending a new REG message upstream towards NextDMC also triggering updates along the path to NextDMC, including the PoA4 and ISR4.

Step(2): NextDMC sends a HREG message to HomeDMC through HomeAS, while also sending a RUPD message to ingress points, which update their FLT with the entry /Prefix::/ISR4.

Step(3): After HomeDMC receives the HREG message, it determines an inter-domain handover for the Producer, and sends CurrentDMC an FREG message with the prefix /CurrentAS:CurrentDMC/FREG. HomeDMC also includes information on the current domain that Producer is attached to CurrentDMC.

Step(4): After CurrentDMC receives the FREG message, it creates a local-scope FREG message and sends it to ISR3. Next, CurrentDMC sends a Route-Update-With-Timeout (RUPDT) message to ingress points requiring them to update their FLT entry associated with /Prefix to point to /NextAS during the FLT-timeout period. Anytime an ingress point receives an Interest targeting /Prefix with the wrong forwarding label during the FLT-timeout interval, the timeout parameter is reset to its default value to ensure that any Consumer thsat is not aware of a domain change is informed.

Step(4.1): The forwarding label for the incorrectly labeled Interest is updated with the correct forwarding label, and the Interest's Mobility-Update tag (MU-tag) is set to 1, before the Interest is forwarded towards the nextAS.

Step(4.2): When Producer receives an Interest with a set MU-tag, suggesting that the Consumer side is not aware of Producer's inter-domain handover, it sets the MU-tag within the Data packet as well, to inform the Consumer side of the inter-domain handover.

Step(4.3): When ISR2 receives a Data packet with set MU-tag, ISR2 re-initiates the Discovery Phase by requesting a forced RUPD from its DMC, which then communicates with HomeDMC to acquire the up-to-date mapping associated with /Prefix. Another alternative to the above approach is for the Producer to include domain information within its Data packets, in response to an Interest with a set MU-tag.

7. Design Discussions

In this section, we address the basic design considerations of a mobility support architecture in an ICN.

7.1. Storage Considerations

FIB represents both the strengths (i.e., flexible operation and on-the-fly routing decisions) and the weaknesses (i.e., overhead) of a CCN/NDN architecture, with greater perceived impact as the network size and content availability increase. Specifically, ad hoc operation allows for greater flexibility during routing, whereas, increased storage requirements (with variable prefix names and multiple entries per prefix) degrade the forwarding efficiency at

each CCN/NDN enabled router by increasing the processing latency [SNDNF].

FLT addresses some of these drawbacks by partitioning the information required for end-to-end packet forwarding at different sections in the network. For instance, prefix-to-locator mappings are stored at the local databases (LDBs) within mainly the DMC nodes. However, because of domain-based partitioning of these namespaces, the number of entries stored at a given LDB is much smaller than what is expected to be stored in the FIB of a CCN/NDN router. Here, ISRs are only responsible for keeping entries of the active hosts they serve, rather than keeping entries for the hosts being serviced by the other routers. Ingress/egress points provide the backbone dependent mappings and their perceived overhead is limited in the maximum of the number of domains and the number of locally hosted Producers within a domain. The intermediate routers, between ISRs and ingress/egress points, are only responsible for carrying the mappings associated with next hop to reach them, hence not anymore observing overhead proportional to the number of hosts being serviced along the downstream channel. As a result, with the proposed architecture, we expect noticeable improvement in the processing latency within the network to route packets between endpoints. Specifically, by forwarding Interests on the controller-managed locator driven address-space, rather than the highly variable prefix-space, lookup latency on a typical CCN/NDN router does not anymore depend on the prefix length.

7.2. Scalability Considerations

For the proposed architecture, another important concern is the performance of the controller node, which is expected to service the requests of the hosts associated with its domain (hosted consumers, or homed entities). A DMC carries prefix-to-domain mappings for the hosted consumers and remote producers and prefix-to-router mappings for the hosted producers. Note that prefix-to-domain mappings are updated at a much lower rate (inter-AS handover rate) than the rate associated with prefix-to-router mappings, which change at the intra-AS handover rate.

7.3. Security Considerations

There are various ways an attacker can exploit the possible vulnerabilities in our architecture by targeting the distributed controllers which can be considered as the entities responsible for managing (authorizing, registering, updating) prefix to locator mappings. For instance, by registering non-existing prefixes to the DMC as fake producers, or by requesting non-existing prefixes from the DMC as fake consumers, attackers can overload the controllers and

limit access to legitimate requests. Following are some of the possible approaches that can be implemented in such cases to minimize the impact of flooding-driven attacks on the overall performance.

7.3.1. Producer Flooding

We assumes a producer to include certain information within the registration message to identify the producer's home network and authenticate the registered content. Hence, we can limit the scope of fake-producer attacks through authentication failure messages received from the home networks. After an authentication failure message is received by the host network's DMC, information on the fake-producer can be shared with the host network's ISRs, to prevent or reduce access to the matching user's registration requests.

7.3.2. Consumer Flooding

To prevent an attacker from hijacking the network by sending requests for non-existent prefixes, multiple approaches are possible. First, we can employ a threshold-based admission policy at the first point of entry for the incoming requests and limit the number of outstanding requests that await for the path update from the DMC. Our architecture already does this to some extent, by suppressing requests targeting the same entity (i.e., Producer) at the ISRs. Second, we can use an adaptive decision policy to enforce stricter threshold values at certain ISRs depending on the experienced overhead at the DMCs. Since the forwarding label in a request message includes information on the entry points, DMCs can aggregate the necessary statistics to quickly determine the problematic areas, and restrict access whenever needed. Third, by sending feedbacks to ISRs on problematic requests, attackers can be identified in a timely manner and the information on them can be shared with other ISRs within the same domain to limit the effectiveness of future attacks.

8. Informative References

[ANCLS] Auge, J., Carofiglio, G., Grassi, G., Muscariello, L., Pau, G., and X. Zeng, "Anchor-less Producer Mobility in ICN", ACM ICN 2015.

[CCN] Jacobson, V., Smetters, D., Thornton, J., Plass, M., Briggs, N., and R. Braynard, "Networking Named Content", ACM CoNEXT, 2009.

[CMOB16] Farahat, H. and H. Hassanein, "Supporting Consumer Mobility Using Proactive Caching in Named Data Networks", IEEE GLOBECOM 2016.

[FWLRP] Azgin, A., Ravindran, R., and G. Wang, "A Scalable Mobility-Centric Architecture for Named Data Networking", IEEE ICCCN Scene Workshop, 2014.

[I-D.ravi-icnrg-ccn-forwarding-label] Ravindran, R., Chakraborti, A., and A. Azgin, "Forwarding Label support in CCN Protocol", draft-ravi-icnrg-ccn-forwarding-label-02 (work in progress), March 2018.

[KITE] Zhang, Y., Zhang, H., and L. Zhang, "Kite: A Mobility Support Scheme for NDN", NDN Technical Report-0020, 2014.

[MAAS] Azgin, A., Ravindran, R., Chakraborti, A., and G. Wang, "Seamless Producer Mobility as a Service in Information Centric Networks", ACM ICN IC5G Workshop, 2016.

[MFRST] Venkataramani, A., Kurose, J., Raychaudhuri, D., Nagaraja, K., Mao, M., and S. Banerjee, "MobilityFirst: A Mobility-centric and Trustworthy Internet Architecture", ACM SIGCOMM CCR, 2014.

[MOB12] Tyson, G., Sastry, N., Rimac, I., Cuevas, R., and A. Mauthe, "A Survey of Mobility in Information-centric Networks: Challenges and Research Directions", ACM MobiHoc Workshop on Emerging Name-Oriented Mobile Network Design, 2012.

[MOB16] Zhang, Y., Afanasyev, A., Burke, J., and L. Zhang, "A Survey of Mobility Support in Named Data Networking", IEEE INFOCOM NOM Workshop, 2016.

[MOBINDN] Zijian, Z., Xiaobin, T., Hebi, L., Zhifan, Z., and M. Dianbo, "MobiNDN: A Mobility Support Architecture for NDN", IEEE CCC 2014.

[NCMP] Adhatarao, S., Chen, J., Arumaithurai, M., Fu, X., and K. Ramakrishnan, "Comparison of Naming Schema in ICN", IEEE LANMAN, 2016.

[NDNMS] Azgin, A., Ravindran, R., and G. Wang, "Mobility Study for Named Data Networking in Wireless Access Networks", IEEE ICC 2014.

[PMNDN] Gao, D., Rao, Y., Foh, C., Zhang, H., and A. Vasilakos, "PMNDN: Proxy Based Mobility Support Approach in Mobile NDN Environment", IEEE Transactions on Network and Service Management, Vol:14, Issue:1, 2017.

[RFC6301] Zhu, Z., Wakikawa, R., and L. Zhang, "A Survey of Mobility Support in the Internet", RFC 6301, DOI 10.17487/RFC6301, July 2011, <https://www.rfc-editor.org/info/rfc6301>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

[SEAML] Ravindran, R., Lo, S., Zhang, X., and G. Wang, "Supporting Seamless Mobility in Named Data Networking", IEEE ICC 2012.

[SNDNF] Yuan, H., Song, T., and P. Crowley, "Scalable NDN Forwarding: Concepts, Issues, and Principles", IEEE ICCCN 2012.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

Aytac Azgin
Huawei Technologies
Santa Clara, CA 95050
USA

Email: aytac.azgin@huawei.com

Ravishankar Ravindran
Huawei Technologies
Santa Clara, CA 95050
USA

Email: ravi.ravindran@huawei.com