icnrg J. Auge, Ed.
Internet-Draft G. Carofiglio
Intended status: Informational L. Muscariello
Expires: May 3, 2020 M. Papalini
Cisco Systems Inc.
October 31, 2019

MAP-Me : Managing Anchorless Mobility in Content Centric Networking
draft-irtf-icnrg-mapme-04

Abstract

Consumer mobility is supported in ICN by design, in virtue of its connectionless pull-based communication model; producer mobility though is not natively supported. This document describes MAP-Me, an anchor-less solution to manage micro-mobility of content producers in the CCN (Content Centric Networking) and NDN (Named Data Networking) architectures, with support for latency-sensitive applications. MAP-Me consists in the combination of two data plane protocols, triggered by producer movements, and leveraging ICN named-based data plane. The main protocol consists in a lightweight FIB update process, complemented by a mechanism of local notification and scoped discovery suitable for low latency applications and fast mobility.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

With the phenomenal spread of portable user devices, mobility has become a basic requirement for almost any communication network as well as a compelling feature to integrate in the next generation networks (5G). The need for a mobility-management paradigm to apply within IP networks has striven a lot of efforts in research and standardization bodies (IETF, 3GPP among others), all resulting in a complex access-dependent set of mechanisms implemented via a dedicated control infrastructure. The complexity and lack of flexibility of such approaches (e.g. Mobile IP) calls for a radically new solution dismantling traditional assumptions like tunneling and anchoring of all mobile communications into the network core. This is particularly important with the increase in rates and mobile nodes (IoT), a vast amount of which never moves.

The Information Centric Network (ICN) paradigm brings native support for mobility, security, and storage within the network architecture, hence emerging as a promising 5G technology candidate. Specifically on mobility management, ICN has the potential to relieve limitations of the existing approaches by leveraging its primary feature, the redefinition of packet forwarding based on "names" rather than "network addresses". Removing the dependence on location identifiers is a first step in the direction of removing the need for any anchoring of communications into fixed network nodes, which may considerably simplify and improve mobility management. Within the ICN paradigm, several architectures have been proposed, as reported in [SURVEY12] and [SURVEY14].

As a direct result of CCN/NDN design principles, consumer mobility is natively supported: a change in physical location for the consumer does not translate into a change in the data plane like for IP. The retransmission of requests for data not yet received by the consumer takes place without involving any signaling to the network. Producer mobility and realtime group communications present more challenges, depending on the frequency of movements, latency requirements, and content lifetime. The topology does not reflect the naming structure, and the mobility management process has to preserve key functionalities such as multipath, caching, etc. In all cases, beyond providing connectivity guarantees, additional transport-level mechanisms might be required to protect the flow performance (see [WLDR] for instance).

MAP-Me aims at tackling such problems by exploiting key CCN/NDN characteristics. Previous attempts have been made in CCN/NDN (and ICN in general) literature to go beyond the traditional IP approaches, by using the existing CCN/NDN request/data packet structures to trace producer movements and to dynamically build a

reverse-forwarding path (see [SURVEY16b] for a survey). They still rely on a stable home address to inform about producer movements or on buffering of incoming requests at the producer's previous point of attachment (PoA), which prevents support for latency-sensitive streaming applications. The approach presented in this document makes a particular focus on this class of applications (e.g. live streaming or videoconferencing) as they have the most stringent performance requirements: negligible per-packet loss-rate and delays. In addition, they typically originate from a single producer and don't allow for the use of caching.

MAP-Me defines a name-based mechanism operating in the forwarding plane and completely removing any anchoring, while aiming at latency minimization. Its performance and guarantees of correctness, stability and bounded stretch are analyzed in [MAPME].

2. MAP-Me overview

2.1. Anchor-less mobility management

Many efforts have been made to define mobility-management models for IP networks in the last two decades, resulting in a variety of complex, often not implemented, proposals. A survey of these approaches is proposed in [RFC6301]. Likewise, within ICN, different approaches to mobility management have been presented [SURVEY13]. Specifically for the CCN/NDN solutions, several surveys of mobility-management approaches can be found [SURVEY16a] [SURVEY16b].

We follow here the classification presented in [MAPME] which highlights their reliance on indirection/rendez-vous points. In particular, a new class of anchor-less approaches is introduced, in which the present proposal fits. Such solutions are less common and have been introduced in ICN to remove the need for anchor points in the data plane, but also in the control plane in the form of resolution or mapping services. These solutions completely remove the use of locators and extend the ICN forwarding mechanisms with mobility support.

2.2. Design principles

o *Micro-Mobility* : MAP-Me addresses micro (e.g. intra Autonomous Systems) producer mobility. Addressing macro-mobility is a non-goal of the proposal. We are focusing here on complementary mechanisms able to provide a fast and lightweight handover, preserving the performance of flows in progress.

  * *Control Plane Agnostic* : MAP-Me _is control-plane agnostic as it does not rely on routing updates or path computation_, which

would be too slow and too costly, but rather works at a faster timescale propagating forwarding updates on a single path. It also leverages real-time notifications left as breadcrumbs by the producer to enable live tracking of its content prefixes and avoid buffering at intermediate nodes. MAP-Me shares the use of data plane mechanisms for ensuring connectivity with [DATAPLANE] which was originally proposed for link failures. This enables the support of high-speed mobility and real-time group applications. In addition, MAP-Me mobility updates are issued at prefix granularity, rather than content or chunk/ packet granularity, to minimize signaling overhead and temporary state kept by in-network nodes, and scale to large and dynamic mobile networks.

o *Access-agnostic* : MAP-Me handles mobility at Layer 3 and is designed to be access-agnostic, to cope with highly heterogeneous wireless access and multi-homed/mobile users.

o *Decentralized and localized* : MAP-Me is designed to be fully _decentralized_, to enhance robustness w.r.t. centralized mobility management proposals subject to single point-of-passage problem. MAP-Me updates are _localized_ and affect a minimum number of routers at the edge of the network to restore connectivity. This effectively realizes traffic off-load close to the end-users.

o *Transparent* : MAP-Me does not involve any name nor modifications to basic request/reply operations to be compatible with standard CCN/NDN design and to avoid issues caused by name modifications like triangular routing, caching degradation, or security vulnerabilities. It does not require consumers or producers to be aware of the mobility of the remote endpoint, nor to perform any handover prediction.

o *Robust* : to network conditions (e.g. routing failure, wireless or congestion losses, and delays), by implementing hop-by-hop retransmissions of mobility updates.

2.3. MAP-Me protocols

As a data plane protocol, MAP-Me handles producer mobility events by means of dynamic FIB updates with the objective of minimizing unreachability of the producer. It relies on the existence of a routing protocol responsible for creating/updating the FIB of all routers, possibly with multipath routes, and for managing network failures (eg. [NLSR]).

MAP-Me is composed of:

- an Update protocol, detailed in Section 3, which is the central component of the proposal;

- a Notification/Discovery protocol, presented in Section 4, which is coupled with the Update protocol to enhance reactivity for realtime/latency-sensitive application, and reduce overhead during fast mobility events.

3. Update protocol

3.1. Rationale

The rationale behind MAP-Me is that the producer announces its movements to the network for all served prefixes, by sending a special Interest packet - named Interest Update (IU) - to "itself" after it reattaches to the network. Such a message looks like a regular Interest packet named with the prefix advertised by the producer. As such, it is forwarded according to the information stored in the FIBs of traversed routers towards all previous locations of the producer known by router FIBs. A special flag carried in the header of the IU enables all routers on the path to identify the Interest as a mobility update and to process it accordingly to update their FIBs (a detailed description of the IU processing is provided in Section 5.3).

The key aspect of the proposal is that it removes the need for a stable home address by directly leveraging name-based forwarding information created by CCN/NDN routing protocols, and eventually further updated due to mobility. FIB updates are triggered by the reception of mobility updates in a fully decentralized way and allow an on-the-fly modification to point to the latest known location of the producer.

3.2. Update propagation

The role of the update process is to quickly restore global reachability of mobile prefixes with low signaling overhead, while introducing a bounded maximum path stretch (the ratio between the selected and the shortest path in terms of hops).

Let us illustrate its behavior through an example where a single producer serving prefix /p moves from position P0 to P1 and so on. Figure 1 (a) shows the initial tree formed by the forwarding paths to the name prefix /p, and on which any IU initiated by the producer will propagate.

```
                               +---+                        +---+
                               | 0 | P0                      | 0 | P0
                               +---+                         +---+
                                ^ ^                           ^ ^
                               /   \                         /   \
                           +---+    +---+                +---+    +---+
                           | 0 |    | 0 |                | 0 |    | 0 |
                           +---+    +---+                +---+    +---+
                            ^ ^                           ^ ^
                           /   \                         /   \
                       +---+    +---+                +---+    +---+
                       | 0 |    | 0 |                | 0 |    | 0 |
                       +---+    +---+             A  +---+    +---+
                        ^ ^                  IU1 /    ^ ^
                       /   \                    /    /   \
                   +---+    +---+          .... .+---+.   +---+
                   | 0 |    | 0 |         .   P1 | 1 | .  | 0 |
                   +---+    +---+        .       +---+ .  +---+
                    ^ ^                  .        ^ ^    .
                   /   \                 .       /   \     .
               +---+    +---+            .   +---+    +---+  .
               | 0 |    | 0 |            .   | 0 |    | 0 |  .
               +---+    +---+            .   +---+    +---+ .
                                          ..................

                        (a)                       (b)

                        Figure 1: IU propagation example
```

```
                                                   .................
                            +---+               ...       +---+     ..
                            | 0 | P0           .          | 1 | P0    .
                            +---+             .        A  +---+       .
                             ^ ^             .  IU(1) /    / ^        .
                            /   \            .       /    V   \       .
                        +---+    +---+      .         +---+    +---+  .
                        | 0 |    | 0 |     .          | 1 |    | 0 |  .
                        +---+    +---+    .        A  +---+    +---+  .
                         ^ ^              . IU(1) /    / ^           .
                        /   \            .       /    V   \         .
            ........+---+.   +---+      .         +---+    +---+   .
           .        | 1 | .  | 0 |     .          | 1 |    | 0 |   .
          .  FIB    +---+ .  +---+    .        A  +---+    +---+   .
         . updated   / ^   .          . IU(1) /    / ^            .
         .          V   \   ....      .      /    V   \          .
         .      +---+    +---+  .     .       +---+    +---+   .
         .  P1  | 1 |    | 0 |  .     .    P1 | 1 |    | 0 |   .
         .      +---+    +---+  .     .       +---+    +---+   .
         .       ^ ^            .     .        ^ ^            .
         .      /   \          .       .      /   \         .
         .  +---+    +---+    .        .  +---+    +---+  .
         .  | 0 |    | 0 |   .         .  | 0 |    | 0 |  .
         .  +---+    +---+  .          .  +---+    +---+  .
          ..................            ..................

                     (a)                       (b)
```

Figure 2: IU propagation example

```
                            +---+
                            | 1 | P1
                            +---+
                           ^  A  ^
                         /    |    \
                    +---+   +---+   +---+
                    | 0 |   | 0 |   | 1 |
                    +---+   +---+   +---+
                                     ^ ^
                                    /   \
                                +---+    +---+
                                | 0 |    | 1 |
                                +---+    +---+
                                          ^ ^
                                         /   \
                                     +---+    +---+
                                 P0  | 1 |    | 0 |
                                     +---+    +---+
                                      ^
                                     /
                                 +---+
                                 | 0 |
                                 +---+
```

Figure 3: IU propagation example

Network FIBs are assumed to be populated with routes towards P0 by a name-based routing protocol. After the relocation of the producer from P0 to P1, once the layer-2 attachment is completed, the producer issues an IU carrying the prefix /p and this is forwarded by the network toward P0 (in general, toward one of its previous locations according to the FIB state of traversed routers).

Figure 1 (b) illustrates the propagation of the IU. As the IU progresses, FIBs at intermediate hops are updated with the ingress face of the IU (Figure 2 (a) and (b)). IU propagation stops when the IU reaches P0 and there is no next hop to forward it to. The result is that the original tree rooted in P0 becomes re-rooted in P1 (Figure 3). Looking at the different connected regions (represented with dotted lines), we see that IU propagation and consequent FIB updates have the effect of extending the newly connected subtree : at every step, an additional router and its predecessors are included in the connected subtree.

3.3. Concurrent updates

Frequent mobility of the producer may lead to the propagation of concurrent updates. To prevent inconsistencies in FIBs, MAP-Me maintains a sequence number at the producer end that is incremented at each handover and associated to all sent IU packets. Network routers also keep track of such sequence number in their FIBs to validate the relative freshness of received updates. The modification of FIB entries is only triggered when the received IU carries a higher sequence number than the locally stored one, while the reception of a less recent update triggers the transmission of a more up-to-date IU backwards in order to fix the not-yet-updated path.

An example reconciliation of concurrent updates is illustrated in Figure 4 (a), when the producer has moved successively to P1 and then to P2 before the first update could complete.

```
                            +---+                         +---+
                            | 0 | P0                      | 2 | P0
                            +---+                      A  +---+
                             ^ ^               IU(2) /    / ^
                            /   \                    /    V   \
                        +---+    +---+                +---+    +---+
                        | 0 |    | 0 |           <!>  | 2 |    | 0 |
                        +---+ A  +---+             A  +---+ A  +---+
                         ^ ^   \            IU(1) /    ^ \   \
                        /   \   \  IU(2)         /    /   V   \ IU(2)
                    +---+    +---+                +---+    +---+
                    | 0 |    | 2 | P2             | 1 |    | 2 |
                 A  +---+    +---+             A  +---+    +---+
          IU(1) /    ^ ^                  IU1 /    / ^
               /    /   \                    /    V   \
                +---+    +---+                +---+    +---+
             P1 | 1 |    | 0 |             P1 | 1 |    | 0 |
                +---+    +---+                +---+    +---+
                 ^ ^                           ^ ^
                /   \                         /   \
            +---+    +---+                +---+    +---+
            | 0 |    | 0 |                | 0 |    | 0 |
            +---+    +---+                +---+    +---+

                     (a)                       (b)
```

Figure 4

```
                            +---+                      +---+
                            | 2 | P0                   | 2 | P2
                            +---+                      +---+
                             / ^                         ^
                            V   \                        |
                        +---+    +---+                 +---+
                    <!> | 2 |    | 2 |                 | 2 |
             IU(2)  /   +---+    +---+                 +---+
                   /     ^ \                           ^   ^
                  V     /   V                         /     \
                    +---+    +---+                +---+     +---+
                    | 2 |    | 2 | P2         P0  | 2 |     | 2 |
          IU(2)  /  +---+    +---+                +---+     +---+
                /    / ^                           ^         ^ ^
               V    V   \                         /    P1   /   \
                +---+    +---+                 +---+    +---+   +---+
             P1 | 1 |    | 0 |                 | 0 |    | 2 |   | 0 |
                +---+    +---+                 +---+    +---+   +---+
                 ^ ^                                     ^ ^
                /   \                                   /   \
            +---+    +---+                          +---+    +---+
            | 0 |    | 0 |                          | 0 |    | 0 |
            +---+    +---+                          +---+    +---+

                     (a)                                 (b)
```

Figure 5

Both updates propagate concurrently until the one with sequence number 1 (IU(1)) crosses a router that has been updated with fresher information. In the example shown in Figure 4 (b), the junction router has already received an IU with higher sequence number (IU(2)). In this case, the router stops the propagation of IU(1) and sends back along its path a new IU with an updated sequence number (Figure 5 (a)). The update proceeds until the whole network has ultimately converged towards P2 (Figure 5 (b)).

MAP-Me protocol reacts at a faster timescale than routing - allowing more frequent and numerous mobility events - and over a localized portion of the network edge between current and previous producer locations. This allows to minimize disconnectivity time and reduce link load, which are the main factors affecting user flow performance, as shown in [MAPME] evaluations.

4. Notification protocol and scoped discovery

IU propagation in the data plane is designed to accelerate forwarding state re-convergence w.r.t. routing or resolution-based approaches operating at control plane, and w.r.t. anchor-based approaches requiring traffic tunneling through an anchor node. Still, network latency makes IU completion not instantaneous and before an update completes, it may happen that a portion of the traffic is forwarded to the previous PoA and dropped because of the absence of a valid output face leading to the producer.

Previous work in the Anchor-Less category has suggested the buffering of Interests at previous producer location to prevent those losses. However, such a solution is not suitable for applications with stringent latency requirements (e.g. real-time) and may be incompatible with IU completion times. Moreover, the negative effects on latency performance might be further exacerbated by IU losses and consequent retransmissions in case of wireless medium. To alleviate such issues, we introduce two enhancements to the previously described behavior, namely (i) an "Interest Notification" mechanism for frequent, yet lightweight, signaling of producer movements to the network and (ii) a scoped "Producer Discovery" mechanism for consumer requests to proactively search for the producer's recently visited locations.

4.1. Interest Notification

An Interest Notification (IN) is a breadcrumb left by producers at every encountered PoA. It looks like a normal Interest packet carrying a special identification flag and a sequence number, like IUs. Both IU and IN share the same sequence number (producers indistinctly increase it for every sent message) and follow the same FIB lookup and update processes. However, unlike IU packets, the trace left by INs at the first hop router does not propagate further. It is rather used by the discovery process to route consumer requests to the producer even before an update process is completed.

It is worth observing that updates and notifications serve the same purpose of informing the network of a producer movement. The IU process restores connectivity and as such has higher latency/ signaling cost than the IN process, due to message propagation. The IN process provides information to track producer movements before update completion when coupled with a scoped discovery. The combination of both IU and IN allows to control the trade-off between protocol reactivity and stability of forwarding re-convergence.

4.2. Scoped discovery

The extension of MAP-Me with notifications relies on a local discovery phase: when a consumer Interest reaches a PoA with no valid output face in the corresponding entry, the Interest is tagged with a "discovery" flag and labeled with the latest sequence number stored in FIB (to avoid loops). From that point on, it is broadcasted with hop limit equal to one to all neighbors and discarded unless it finds a breadcrumb left by the producer with a higher sequence number. The notifications can either allow to forward consumer Interests directly to the producer or give rise to a repeated broadcast in case of no valid output face. The latter is the case of a breadcrumb left by the producer with no associated forwarding information because the producer has already left that PoA as well. A detailed description of the process is reported in Section 5.3.

The notification/discovery mechanism proves important to preserve the performance of flows in progress, especially when latency-sensitive.

4.3. Full approach

The full MAP-Me approach consists in the combination of Updates and Notifications through a heuristic allowing the producer or its PoA to select which type of packet to send. One such heuristic consist in sending a IN immediately after an attachment and a IU at most every Tu seconds, which allows to reduce signaling overhead during periods of high-mobility. The Tu parameter allows to tune the timescale at which Updates occur, and leads to a trade-off between signaling and discovery overhead [MAPME]. The definition of more advanced heuristics is out of scope for the present draft.

5. Implementation

In this section we describe the changes to a regular CCN/NDN architecture required to implement MAP-ME and detail the above-described algorithms. This requires to specify a special Interest message, additional temporary information associated to the FIB entry and additional operations to update such entry.

5.1. MAP-Me messages

MAP-Me signaling messages are carried within user plane as special Interest messages corresponding to "update" and "notification", and their corresponding acknowledgements.

Two new optional fields are introduced in a CCN/NDN Interest header:

- an "Interest Type" (T) used to specify one of the four types of messages: Interest Update (IU), Interest Notification (IN), and as well as their associated acknowledgment (Ack) messages (IU_Ack and IN_Ack). Those flags are recognized by the forwarding pipeline to trigger special treatment;

- a "sequence number" to handle concurrent updates and prevent forwarding loops during signaling, and to control discovery Interests' propagation;

5.2. Data structures and temporary state

FIB entries are augmented with information required for mobility management, that we denote as Transient FIB buffer, or simply TFIB, and sketch in Figure 6:

- a "sequence number" which is incremented upon reception of IU/IN messages. It can be assumed this counter is set to 0 by the routing protocol.

- a list of so-called "previous next hop(s)" (further denoted as PrevHops), similar to the list of NextHops in the original FIB, which temporarily stores information about faces that were previously next hops, and should still be memorized to allow for retransmissions and thus ensure the consistency of MAP-Me operations. They typically correspond to nodes for which an IU has been sent, but no acknowledgement (ACK) has yet been received (upon which they are cleared). In case of notifications, no ACK is expected, and those entries serve as a memory of the former tree structure that will be restored upon producer departure. We flag those entries with a boolean marker indicating if they correspond to an IU (and thus should be monitored for retransmissions) or an IN (in which case they just serve as memory for further use).

```
      IU (IN) input face(s)           IU (IN) output face

+-----------+-------------------+.......+..........................+
|  /prefix  |  { next hop(s) }  |  seq  | { previous next hop(s) } |
+-----------+-------------------+.......+..........................+
 \_____________ _______________/ \_______________ _______________/
               V                                  V
          original FIB                       TFIB section
```

Figure 6: MAP-Me FIB/TFIB description

5.3. Algorithm description

5.3.1. Producer attachment and face creation

MAP-Me operations are triggered by a change of adjacencies in the network, reflected in the forwarder by the creation or removal of a face. This can be for instance the layer 2 detachment and attachment following a mobility/handover event, but also any other mechanism such as point-to-point IP link or UDP tunnel for instance, as allowed by the forwarder implementation.

One realization of this architecture is to delegate face management to a third party agent, keeping the ICN forwarder state synchronized with the underlying topology, and having MAP-Me only react to changes in the face table.

5.3.2. IU/IN transmission at producer

The creation of a new face on the producer triggers the increase of MAP-Me sequence number and the transmission for every locally served prefix, of an IU or IN carrying the updated sequence number.

5.3.3. IU/IN transmission at network routers

At the reception of IU/IN packets, each router performs a name-based Longest Prefix Match lookup in FIB to compare sequence number from IU/IN and from FIB. According to that comparison:

- if the IU/IN packet carries a higher sequence number, the existing next hops associated to the lower sequence number in FIB are used to forward further the IU (INs are not propagated) and temporarily copied into TFIB to avoid loss of such information before completion of the IU/IN acknowledgement process. The ingress face of the IU/IN is then added to FIB to route consumer requests to the latest known location of the producer.

- If the IU/IN packet carries the same sequence number as in the FIB, the originating face of the IU/IN is added to the existing ones in FIB without additional packet processing or propagation. This may occur in presence of multiple forwarding paths.

- If the IU/IN packet carries a lower sequence number than the one in the FIB, FIB entry is not updated as it already stores 'fresher information'. To advertise the latest update through the path followed by the IU/IN packet, this one is re-sent through the originating face after having updated its sequence number with the value stored in FIB.

The operations in the forwarding pipeline for IU/IN processing are reported in Figure 7, where we make use of the following primitives: - Send(Interest, Face) is used to send the specified Interest on the specified Face. - ProcessTFIB() sends an IU for all flagged entries in the TFIB, using the latest sequence number stored in the FIB entry, and schedule the entry to be checked for retransmissions.

```
| Algorithm 1:ForwardSpecialInterest(SpecialInterest SI,IngressFace F)
|
|  CheckValidity()
|  // Acknowledge reception
|  s <- e.seq
|  e.seq <- SI.seq
|  Send(IU_Ack(e.seq), F)
   flag <- (SI.type == IU)
|  // Retrieve the FIB entry associated to the prefix
|  e <- FIB.LongestPrefixMatch(SI.name)
|  if SI.seq >= e.seq then
|  .   //Process special interest
|  .   e.TFIB = e.TFIB \ { F }
|  .   if SI.seq > s then
|  .   .   e.TFIB = e.TFIB U { (f, flag) | f in (e.NextHops \ F) }
   .   .   ProcessTFIB()
|  .   .   e.NextHops = {}
|  .   e.NextHops = e.NextHops U { F }
|  else
|  .   // Send updated IU backwards
|  .   SI.seq = e.seq
|  .   e.TFIB = e.TFIB U { (F, flag) }
|  .   ProcessTFIB()
```

Figure 7

5.3.4. Reliable transmission

MAP-Me ensure the reliable delivery of signaling messages thanks to a retransmission timer which reissue Interest Updates (eventually carrying updated sequence number as found in the FIB), if no corresponding ACK has been received in a predefined interval, and whose sequence number has to match the one stored in the FIB.

A slotted implementation of such scheme is possible by using a single timer, and keeping a list of FIB entries that require to be checked for pending retransmissions in the next slot. Upon timer expiration, if all required ACKs have been received, the TFIB will be empty and the entry does not have to be tracked anymore. Otherwise, necessary retransmissions are performed and the entry will be checked again in

the next slot. When no entry has to be monitored, the process can sleep until the next mobility event.

5.3.5. Consumer request forwarding in case of producer discovery

The forwarding of regular Interests is mostly unaffected in MAP-Me, except in the case of discovery Interests that we detail in Figure 8. The function SendToNeighbors(I) is responsible for broadcasting the Interest I to all neighboring PoAs.

```
| Algorithm 2:  InterestForward(Interest I, Origin face F)
|
|  // Regular PIT and CS lookup
|  e <- FIB.LongestPrefixMatch(I.name)
|  if e = 0 then
|  .   return
|  if I.seq = 0 then
|  .   // Regular interest
|  .   if hasValidFace(e.NextHops) or DiscoveryDisabled then
|  .   .   ForwardingStrategy.process(I, e)
|  .   else
|  .   .   // Enter discovery mode
|  .   .   I.seq <- e.seq
|  .   .   SendToNeighbors(I)
|  else
|  .   // Discovery interest: forward if producer is connnected
|  .   if hasProducerFace(e.NextHops) then
|  .   .   ForwardingStrategy.process(I, e)
|  .   // Otherwise iterate iif higher seq and breadcrumb
|  .   else if e.seq >= I.seq and EXISTS f |(f -> NULL) in e.TFIB then
|  .   .   I.seq <- e.seq
|  .   .   SendToNeighbors(I)
```

Figure 8

When an Interest arrives to a PoA which has no valid next hop for it (because the producer left and the face got destroyed), it enters a discovery phase where the Interest is flagged as a Discovery Interest and with the local sequence number, then broadcasted to neighboring PoAs.

Upon reception of a Discovery Interest, the PoA forwards it directly to the producer if still attached, otherwise it repeats the one-hop brodcast discovery to neighboring PoAs if it stores a recent notification of the producer presence, i.e. an entry in TFIB having higher sequence number than the one in the Discovery Interest. Otherwise, the Discovery Interest is discarded.

It is worth observing that the discovery process is initiated only in the case of no valid next hop, and not every time a notification is found in a router. This is important to guarantee that the notification/discovery process does not affect IU propagation and completion.

5.3.6. Producer departure and face destruction

Upon producer departures from a PoA, the corresponding face is destroyed. If this leads to the removal of the last next hop, then faces in TFIB corresponding to IN are restored as next hops in the FIB so as to preserve the original forwarding tree and thus global connectivity.

6. Security considerations

All mobility management protocols share the same critical need for securing their control messages which have a direct impact on the forwarding of users' traffic. [SEC] reviews standard approaches from the literature and proposes a fast, lightweight and decentralized approach based on hash chains that can be applied to MAP-Me and fits its design principles.

7. Acknowledgements

The authors would like to thank Giulio Grassi (UPMC/UCLA), Giovanni Pau (UPMC/UCLA) and Xuan Zeng (UPMC/SystemX) for their contribution to the work that has led to this document.

8. IANA Considerations

This memo includes no request to IANA.

9. References

9.1. Normative References

[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>.

9.2. Informative References

[DATAPLANE]
J, ., A, ., A, ., B, ., M, ., and . S, "Ensuring connectivity via data plane mechanisms.", 2013.

[KITE] Zhang, Y., Zhang, H., and L. Zhang, "Kite", Proceedings of the 1st international conference on Information-centric networking - INC '14, DOI 10.1145/2660129.2660159, 2014.

[MAPME] Auge, J., Carofiglio, G., Grassi, G., Muscariello, L., Pau, G., and X. Zeng, "MAP-Me: Managing Anchor-Less Producer Mobility in Content-Centric Networks", IEEE Transactions on Network and Service Management Vol. 15, pp. 596-610, DOI 10.1109/tnsm.2018.2796720, June 2018.

[NLSR] Hoque, A., Amin, S., Alyyan, A., Zhang, B., Zhang, L., and L. Wang, "NISR", Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking - ICN '13, DOI 10.1145/2491224.2491231, 2013.

[SEC] Compagno, A., Zeng, X., Muscariello, L., Carofiglio, G., and J. Auge, "Secure producer mobility in information-centric network", Proceedings of the 4th ACM Conference on Information-Centric Networking - ICN '17, DOI 10.1145/3125719.3125725, 2017.

[SURVEY12] Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., and B. Ohlman, "A survey of information-centric networking", IEEE Communications Magazine Vol. 50, pp. 26-36, DOI 10.1109/mcom.2012.6231276, July 2012.

[SURVEY13] Tyson, G., Sastry, N., Rimac, I., Cuevas, R., and A. Mauthe, "A survey of mobility in information-centric networks", Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications - NoM '12, DOI 10.1145/2248361.2248363, 2012.

[SURVEY14] Xylomenos, G., Ververidis, C., Siris, V., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K., and G. Polyzos, "A Survey of Information-Centric Networking Research", IEEE Communications Surveys & Tutorials Vol. 16, pp. 1024-1049, DOI 10.1109/surv.2013.070813.00063, 2014.

[SURVEY16a] Feng, B., Zhou, H., and Q. Xu, "Mobility support in Named Data Networking: a survey", EURASIP Journal on Wireless Communications and Networking Vol. 2016, DOI 10.1186/s13638-016-0715-0, September 2016.

[SURVEY16b]
Zhang, Y., Afanasyev, A., Burke, J., and L. Zhang, "A survey of mobility support in Named Data Networking", 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), DOI 10.1109/infcomw.2016.7562050, April 2016.

[WLDR] Carofiglio, G., Muscariello, L., Papalini, M., Rozhnova, N., and X. Zeng, "Leveraging ICN In-network Control for Loss Detection and Recovery in Wireless Mobile networks", Proceedings of the 2016 conference on 3rd ACM Conference on Information-Centric Networking - ACM-ICN '16, DOI 10.1145/2984356.2984361, 2016.

Authors' Addresses

Jordan Auge (editor)
Cisco Systems Inc.
11, rue Camille Desmoulins
Issy-les-Moulineaux 92130
France

Email: augjorda@cisco.com

Giovanna Carofiglio
Cisco Systems Inc.
11, rue Camille Desmoulins
Issy-les-Moulineaux 92130
France

Email: gcarofig@cisco.com

Luca Muscariello
Cisco Systems Inc.
11, rue Camille Desmoulins
Issy-les-Moulineaux 92130
France

Email: lumuscar@cisco.com

Michele Papalini
Cisco Systems Inc.
11, rue Camille Desmoulins
Issy-les-Moulineaux 92130
France

Email: micpapal@cisco.com