draft-forwarding-label-ccn-02.txt

Ravi Ravindran, Asit Chakraborti, Aytac Azgin

Huawei

(IETF/ICNRG-95, Buenos Aires)

[ravi.ravindran@huawei.com]

[asit.chakraborti@huawei.com]

[aytac.azgin@huawei.com]

Draft ToC

- ID-locator Namespace Split in CCN
- FL-Object Proposal
  - FL Object Naming/Insertion/Swapping/Termination
- FL Object Message Format
- FL Object Processing Rules
- PIT Processing Implications
- Multi-Domain Considerations
- FL Object Security
- Use case scenarios
  - Handling Producer Mobility
  - Manifests
  - Interest Routing Optimization
  - Routing Scalability

Draft Objectives

- Third iteration of this draft.
- Proposes to have ID/Locator namespace Split in CCN.
  - The locator is called **Forwarding-Label Object** which can be modified in the infrastructure.
- Could be used serve different objectives:
  - Routing Scalability
  - Producer Mobility
  - Opportunistic Indirections (off-path caching)
  - Service Affinity (Edge Computing)
  - In-Network Computing (e.g. NFN)
  - Inter-domain Routing
  - ..
- We provide details on using Forwarding Label for producer mobility
- We discuss the use of Forwarding Label for Manifests to support Flat-IDs (e.g. ContentObjectHashId) and Routeable Names.
  - Provide a logic to handle this kind of forwarding.

Definitions

- **Identifier (ID)** is a persistent secure or non-secure flat-ID or a hierarchical name assigned to a content, device or service. If the ID is secure, then trust relationship can be derived from it. **Generally the identifier space is managed by applications**.
- **Locator (LID)** is a routeable topological name assigned to a network entity such as a router, a server, or an end device. **Generally the locator space is managed and assigned by the network administrators.**

FL Object Proposal

- Allows insertion of locator names in the fixed header of the Interest message.
- Insertion can be by applications or by the network.
- FL Object management is governed by usage and its policy, basic operations include
  - FL Insertion : Mapping ID to LID
  - FL Swapping : one LID can be swapped with another LID
  - FL Termination: once a LID is reached, ID based routing will follow
- Security Concerns
  - Security related to the ID-LID mapping system
  - Malicious applications poisoning cache resources

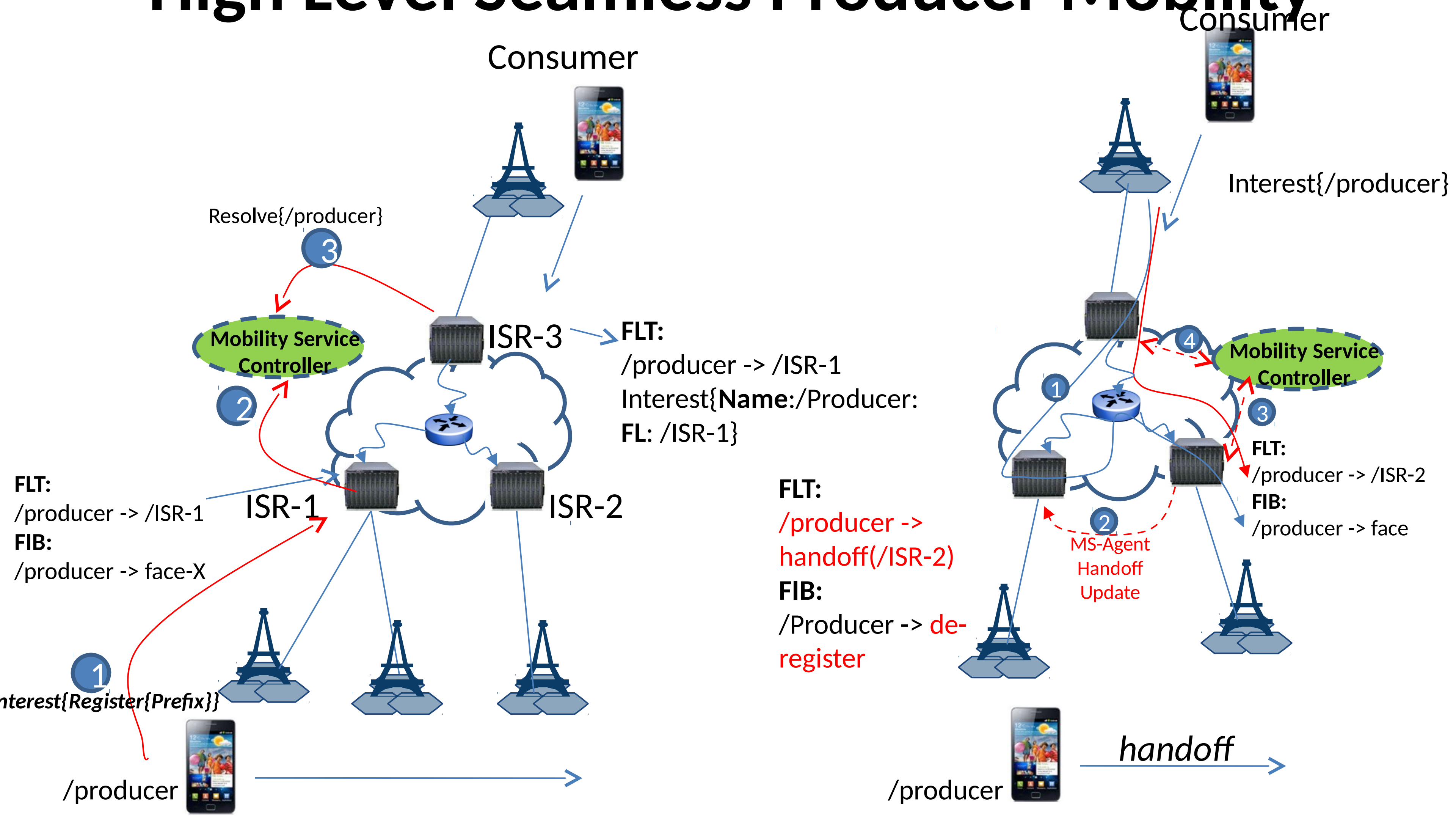
Use Case 1: Forwarding Label for Producer Mobility

- Two kinds of mobility solutions proposed so far:
  - Application-based Approach:
    - Here Applications actively initiates state change in the network to enable Interest routing to its current location, e.g. Kite (tracing/traced Interests) and Anchorless proposals (Interest Update/Notification)
      - Pros: Still allows to work on a single namespace
      - Cons : Scalability, Security concerns with increasing number of mobile producers
      - How about Flat-ID, e.g. IoT ?
  - Network-based Approach
    - Relies on routing on two name space, one is the application namespace and network namespace (forwarding-label).
    - Binding happens through a registration process, this may only be used for mobile or non-routeable names.
    - A solution for using forwarding label was presented **"Mobility as a Service in CCN**", IETF/ICNRG Paris Meeting, 2016 https://www.ietf.org/proceedings/interim/2016/01/14/icnrg/slides/slides-interim-2016-icnrg-1-10.pdf

Producer Mobility Using FL

1. Producer registers the name prefix that requires mobility with the edge ISR (ICN Service Router, more functions such as flow classification, FL Cache table)
2. The ISR registers the name and location with a mobility service controller (MSC).
3. When Interests arrive at the ISR, FL object is appended to the fixed header.
4. At the FL destination, the FL is removed and name-based routing is conducted.
    1. If the name based routing indicates the producer has moved, and a new FL object is present, the Interest header is updated and forwarded.
    2. The path stretch from the ingress PoA to the current egress PoA is addressed using in-band marking of the ‘U’ turned Interests and corresponding Content-Objects.
5. The new PoA eventually updates the MSC about its new location.

High Level Seamless Producer Mobility



FLT:
/producer -> /ISR-1
Interest{**Name**:/Producer:
FL: /ISR-1}

FLT:
/producer -> /ISR-1
FIB:
/producer -> face-X

FLT:
/producer -> handoff(/ISR-2)
FIB:
/Producer -> de-register

FLT:
/producer -> /ISR-2
FIB:
/producer -> face

- **Seamless mobility is handled by the ISRs in the edges.**
- **The ‘U’ turned Interests are marked, so that these Interests leave no new trace.**
- **The returned Content Object are also marked by the ISR, triggering update by the ingress ISR, thus handling the path strech**
- **The mobility state change in the ISR to aid mobility can be UE or Network driven.**

FL for Manifests

- FL can be used for support retrieval for nameless objects (or other secure flatID), with out overloading the definition of name that was originally intended, i.e. of Content Names [1].
- Current proposal [2], ContentObjectHashID is used as Hash Restriction parameter, and Locator as a name.
- These are the potential issues with this approach
  - Forwarder doesn't have a direct means to distinguish a content name from the locator in the Interest for more meaningful processing.
  - All Interests have to be first checked for ContentObjectHashID, which is used for indexing in the PIT, instead of Interest name all the time.
  - Situations may arise where a producer may include two IDs, a content name along with the ContentObjectHashID, in which case the content name may be preferred to avoid hash computation.
    - This situation has to be avoided.

[1] Van Jacobson et al, "Networking Named Content", ACM, CoNext, 2009
[2] Mosko, M., "Nameless Objects.", IETF/ICNRG, Paris Interim, 2016

FL for Manifests

- These issues can be avoided by using ContentObjectHashID as a name and using Locator in the FL Object.
- For current CCNx1.0, proposal, a new name type can be defined to identify Flat-IDs (e.g. ContentObjectHashID) differentiating it from the routeable names.
  - Hence forwarder admits Interests with FL object with only ContentObjectHashID, to avoid cache poisoning.
- We propose a high level forwarding logic for this scenario.

High Level Forwarding Logic

```
Begin
  - If Edge_Router (or Gateway)
      If Interest arrives on a face with a flat-ID (type ContentObjectHashID)
          Then check for the presence of FL object
              If FL object is present, use the LID in the FL object for Interest forwarding
          If there no FL Object
              If policy allows, resolve the flat-ID with a NRS to obtain an FL object
                  Use the FL object to route the Interest
      End
      If the Interest arrives with a routeable ID
          If there is no FL object
              Match Interest ID with name policy
                  If a name policy for resolution exists
          Resolve ID-> LID mapping, FL Object
            Use the FL object for routing
        If there is FL Object
            Then use the ID for forwarding, Remove the FL object
        End
    End
  - if Core_Router
      if Interest arrives with a flat-ID
        Use the FL Object for forwarding
       Else if Interest is with a Routeable ID
         Use the name for forwarding
    End
End
```

- Ensures, applications only use ContentObjectHashID, and networks use FL-Object to server different services in the network

Next Steps

- Comments

Thank You and Questions