Scalable Container-based Routing Scheme for ICN

2014-02-25

Joo-Chul Kevin Lee

Background

- ICN (Information Centric Networking)
  - Increasing demand for scalable and efficient distribution of contents
  - Moving focus from “host-oriented” to “information object-oriented”
  - New network architecture focusing on “information objects”

Challenges

- Naming scheme
  - Scalable name-based routing
- Scalable routing & resolution system
  - Route-By-Name Routing (RBNR)
  - Lookup-By-Name Routing (LBNR)
- In-network caching
  - Authentication
  - Synchronization
- Mobility
- Security

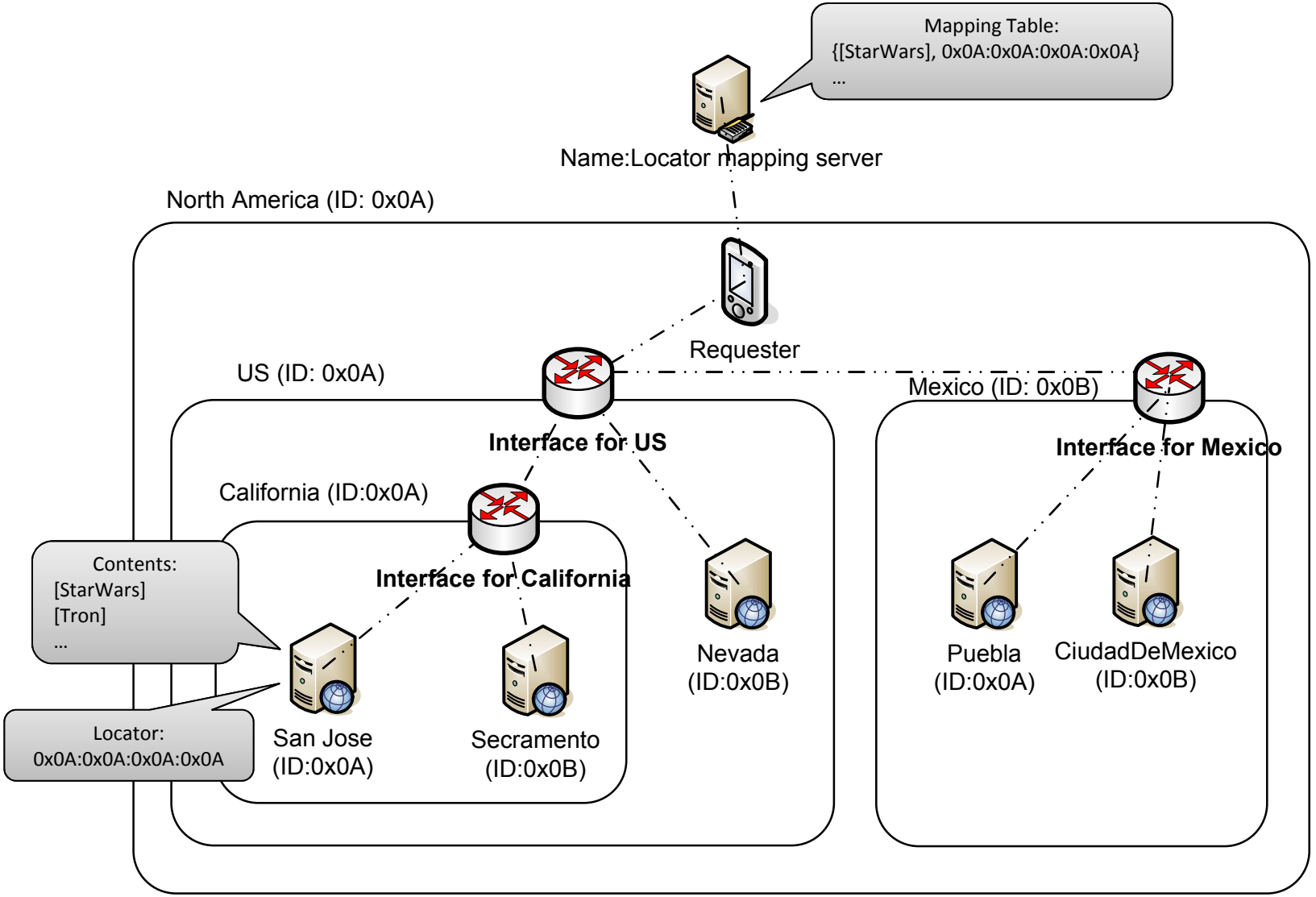
Motivation

- We are thinking...
  - More aggregation in routing system
  - Something that helps us to guess closer cached object
- Thus, we are proposing...
  - Hierarchical container architecture for routing system
- We are assuming...
  - “flat name”
  - Lookup-By-Name Routing (LBNR) case

Basic Concepts

- Container
  - A container is ...
    a “contents server”
    or
    a group of “contents servers” and/or “other sub-container”
  - A container in higher tier looks like a “contents server” from outside
- Locator
  - Described as a concatenation of “Container ID”
  - Indicates a relative-position in “container-space”
  - Can be used as “*routing hint*”
  - Content server registers “locator” for the contents that it contains (name resolving system)
- Container interface
  - A border router of a container
  - Maintains routing table whose key is “locator”

Example Topology



Example Topology

- Container
  - North America (ID: 0x0A)
  - US (ID: 0x0A)
  - California (ID: 0x0A)
  - San Jose (ID: 0x0A)
  - Sacramento (ID: 0x0B)
  - Nevada (ID: 0x0B)
- Locators
  - San Jose ➔ North America:US:California:San Jose (0x0A:0x0A:0x0A:0x0A)
  - Sacramento ➔ North America:US:California:Sacramento (0x0A:0x0A:0x0A:0x0B)
  - Nevada ➔ North America:US:Nevada (0x0A:0x0A:0x0B)

Procedure :: Name Resolution

- Name resolution
  - Name resolution depends on external “name resolution system”
  - Implementation of name resolution system is out of scope
    - We are considering hierarchical system adopting bloom filter technique

Procedure :: Discovery

- Request Message

IP header **(SRC: requestor's IP, DST: Container Interface's IP address)**
Content name (e.g. self-certifying ID)
Content locator

After name resolution

IP header **(SRC: requestor's IP, DST: Interface #1's IP address)**
"StarWars"
0x0A:0x0A:0x0A:0x0A

Procedure :: Discovery

- Forwards Request message
  - Each container interface forwards request message according to "locator" embedded in the request message

Interface for US's routing table

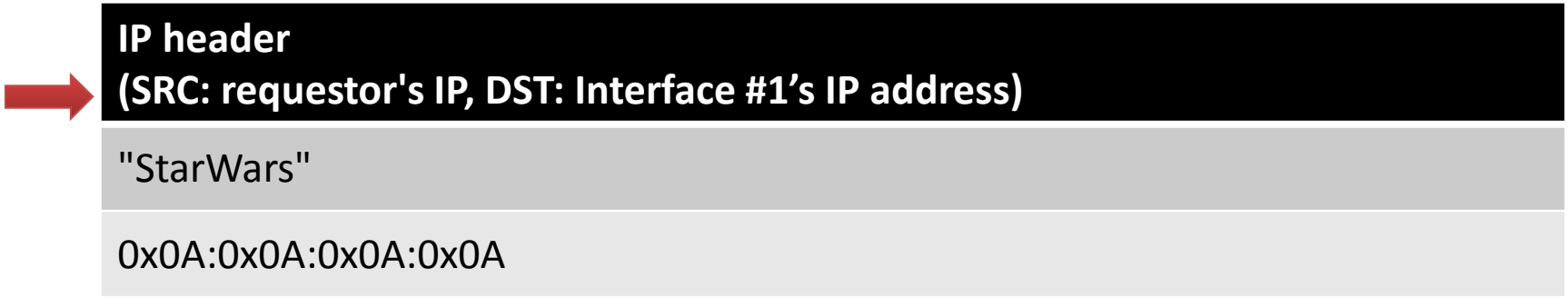
Destination	Nexthop	Out if
0x0A:0x0B	Interface for Mexico's IP	If 0
0x0A:0x0A:0x0A	**Interface for California's IP**	**If 1**
0x0A:0x0A:0x0B	Nevada's IP	If 2

Interface for California's routing

Destination	Nexthop	Out if
0x0A:0x0B	Interface for US's IP	If 0
0x0A:0x0A:0x0B	Interface for US's IP	If 0
0x0A:0x0A:0x0A:0x0A	**San Jose's IP**	**If 1**
0x0A:0x0A:0x0A:0x0B	Sacramento's IP	If 2

Procedure :: Delivery

- Delivery requested content to the requester
  - Sends contents to the source IP of request message (direct way)

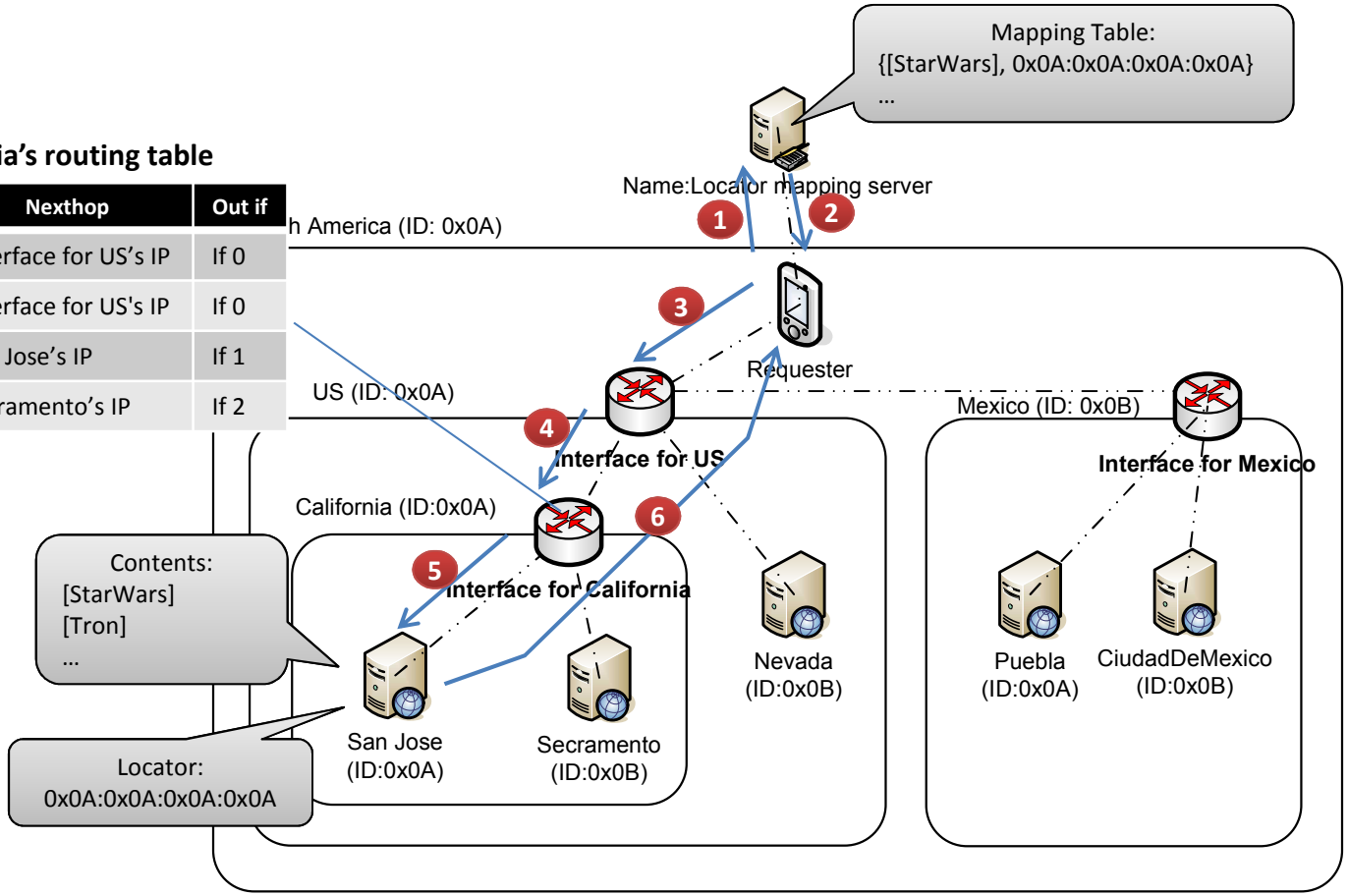


  - Sends contents to the "locator" of requestor (reverse path)
    - Could make cached contents

Example Topology

Interface for California's routing table

Destination	Nexthop	Out if
0x0A:0x0B	Interface for US's IP	If 0
0x0A:0x0A:0x0B	Interface for US's IP	If 0
0x0A:0x0A:0x0A:0x0A	San Jose's IP	If 1
0x0A:0x0A:0x0A:0x0B	Sacramento's IP	If 2



Build routing table

- Build routing table for “container interface”
  - Each container interface runs “modified link-state routing protocol”
  - Each container interface forwards LSA to the other container according to the following filtering rules to reduce topology graph
    - LSAs in tier N container are forwarded to all container interfaces which belong to tier N
    - LSAs in tier N-1 are injected to tier N
    - LSAs in tier N+1 are filtered, LSA which only includes aggregated locator is injected to tier N instead.

In-network Cache

- In-network caching
  - Contents can be cached on each container interface
  - Container interface adds current locator to the mapping entry of original content name
    - E.g. { *name*, *locator1*, *locator2*, ... }
  - User can decide closer cached content by comparing its current locator with locator(s) mapped to the content name (by **Longest matching**)

Summary

- We proposes scalable container-based routing scheme for ICN
- Hierarchical container structure abstracts contents servers
- Hierarchically organized container structure makes routing system scalable
  - Concatenation of container IDs plays role of “locator”
  - Locator information is propagated through routing protocol
- In-network caching is easily implemented on this architecture