Bloom Filter-based Flat Name Resolution System for ICN

IEFT 90

Jungha Hong, Woojik Chun, and Heeyoung Jung

July 22, 2014

Background

- Name is assigned to data object
  - *Location independent and flat name* is assumed

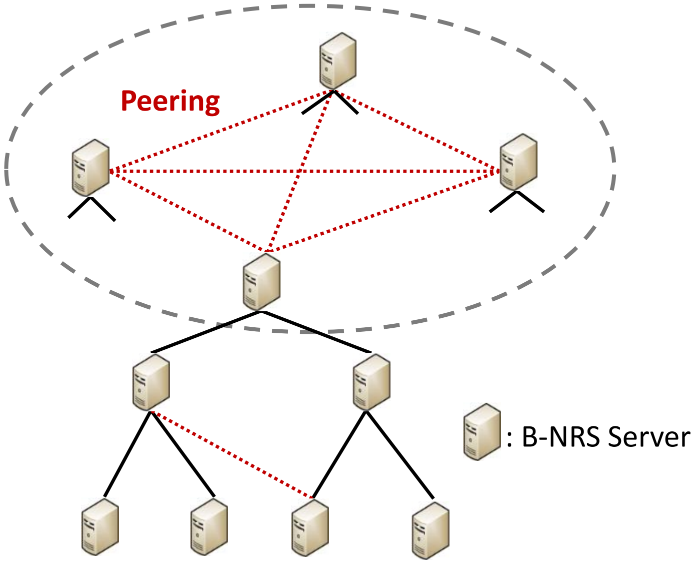
- The binding between name and locator is required for lookup-by-name routing
  - *Name resolution system (NRS)*
    - Maintains and resolves the bindings
    - Remarkably, it is required for *ICN-IoT architecture* (draft-zhang-iot-icn-architecture-01)

- The most important challenge on designing NRS is *scalability* on the ever-increasing number of named data object (NDO)
  - *Hierarchical structure of NRS and bloom filter* are utilized
    - Constructs a network of NRS servers, which consists of a forest by several disjoint trees
    - Bloom filter as an aggregated form of names is announced instead of announcing the whole list of names

Bloom filter-based NRS (B-NRS)

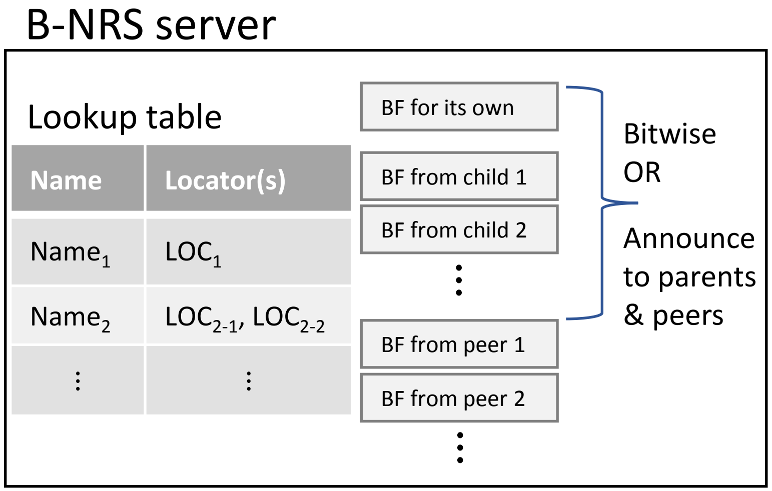
B-NRS structure

- A network of B-NRS servers
- Relationships of parent-child and peering



B-NRS server components

- Name lookup table
- Bloom filters (BFs) for its own, from child, and from peer



B-NRS server

Lookup table

Name	Locator(s)
Name_1	LOC_1
Name_2	$\text{LOC}_{2\text{-}1}$, $\text{LOC}_{2\text{-}2}$
⋮	⋮

BF for its own

BF from child 1

BF from child 2

⋮

BF from peer 1

BF from peer 2

⋮

Bitwise OR

Announce to parents & peers

Key operations

- Name registration
  - *No constraints*
    - Flat names can be registered to any arbitrary B-NRS server
  - Followed by BF updates : *insert-through*
  - No BF update when name is deleted : use *periodic refresh*
    - BF cannot handle deletion
- Locator update
  - When a NDO presents/depresents into/from the network, only locator information is inserted/deleted into/from lookup table
    - No effects on BFs and structure of B-NRS
    - *Inherently supports mobility*
- Lookup
  - Through the BF test with the given name, locator lookup request is forwarded into the B-NRS server where the binding is actually stored
    - Lookup request is forwarded into all child and peer severs which return positive answers
    - If none of BFs returns positive answer, then lookup request is forwarded into parent server
  - Locator lookup reply takes the reverse path of the lookup request

Why bloom filter? Why not DHT?

- DHT-based approach
  - One popular scalable technique to distribute flat names
  - Some drawbacks such as *ownership, locality, deployment*, etc.
    - We utilize BF to overcome such drawbacks of DHT
- Two major benefits of BF
  - *Fixed constant time of insertion and search*
    - Completely independent of the number of elements in the set
  - *Efficient support for union of BFs* with the same size and set of hash functions
    - It can be implemented with bitwise OR
- Drawbacks of BF
  - False positive
    - *Can be minimized* by choosing the optimal parameters
  - No member deletion
    - Taken care by *periodic reconstruct of the BFs*
- Complexity
  - DHT-based approach : $log_2 n$ where n is the total number of NRS servers
  - B-NRS : $log_d n$ where d is the degree of B-NRS servers in trees

Thank you

www.idnet.re.kr