Bloom Filter-based Flat Name Resolution System for ICN

IEFT 91, Honolulu

Jungha Hong, Woojik Chun, and Heeyoung Jung

November 10, 2014

Problem and challenge

- *Location independent and flat name* is assumed in ICN
- The binding between name and locator is required for lookup-by-name routing
- Name resolution system (NRS) maintains and resolves the bindings
- The most important challenge on designing NRS is *scalability* on the ever-increasing number of named data object

Bloom filter-based NRS (B-NRS)

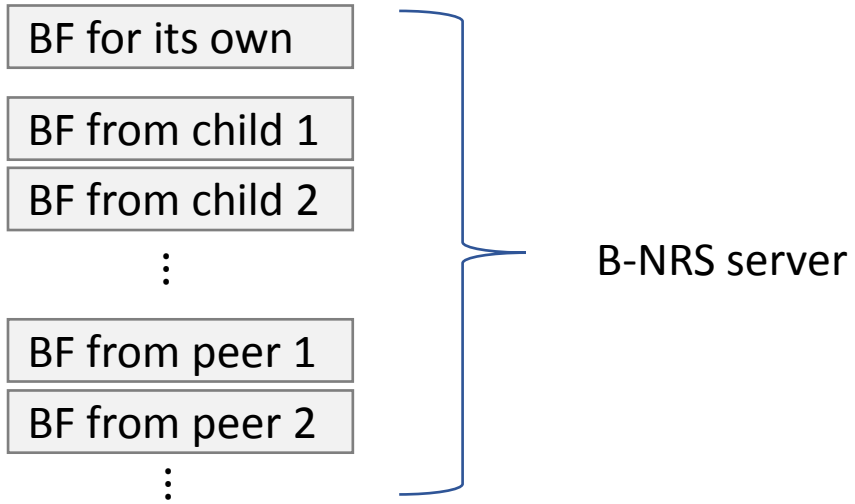
- Hierarchical structure
  - Constructs a network of B-NRS servers
  - B-NRS consists of a forest by several disjoint trees
  - Relationships of parent-child and peering
    - No constraint for peering
- Bloom filter (BF)
  - Used as an aggregated form of names
  - Announced instead of the whole list of names

B-NRS server components

- Name lookup table

Name	Locator(s)
$Name_1$	LOC_1
$Name_2$	$LOC_{2\text{-}1}$, $LOC_{2\text{-}2}$
$Name_3$	-
⋮	⋮

- Bloom filter



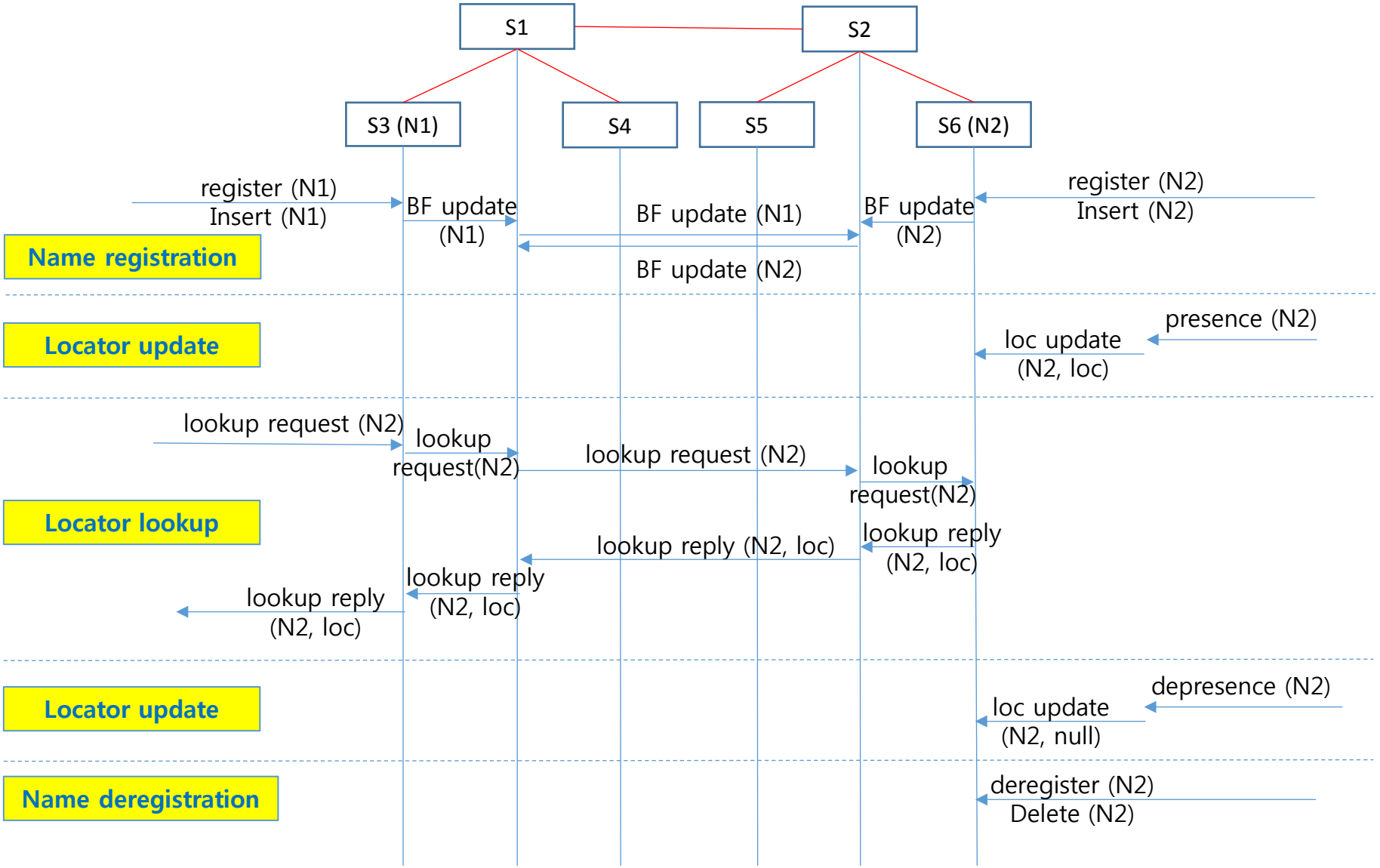
Key operations (1)

- 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
    - BF cannot handle deletion
- Locator update
  - Locator information for the registered names is inserted(deleted) into(from) name lookup table
    - No effects on both BFs and structure of B-NRS
    - *Inherently supports mobility*

Key operations (2)

- Locator 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 from the BF test
    - 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

Operation flows

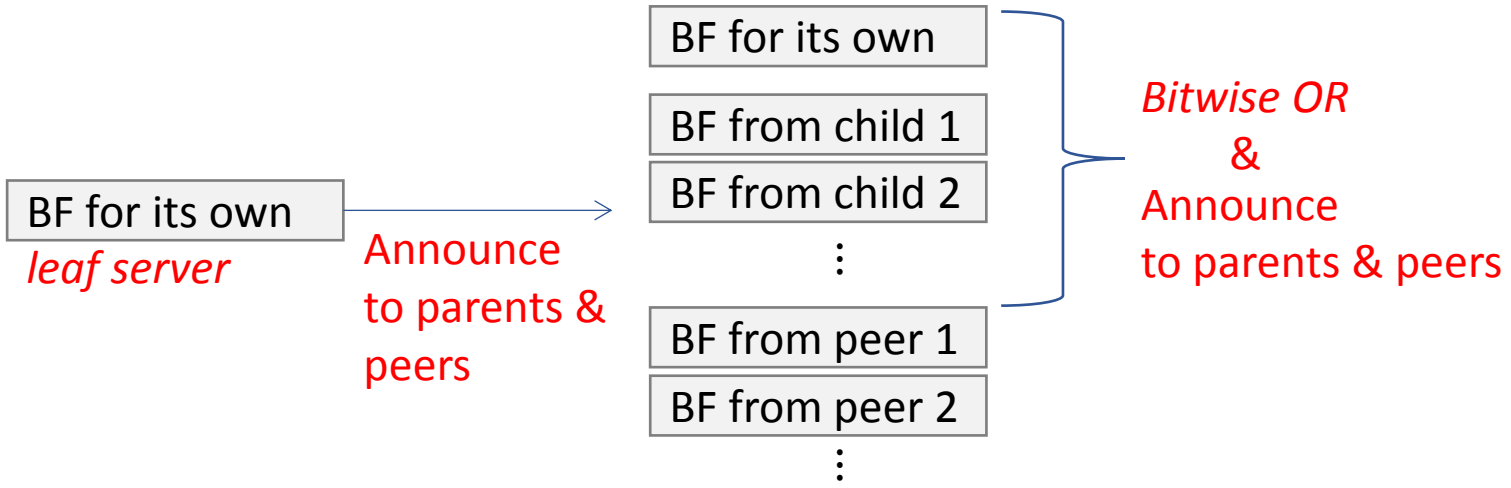


How to handle BF deletion issue

- Completely ignoring the BF updates for deletion
  - B-NRS would eventually be able to find the locator for a given name
  - Generates the fewest number of control messages for BF updates but significantly increases the false positive rates
- Updates the entire BFs in B-NRS for every single deregistration
  - Generates the lowest false positive rates but extremely high control message for BF updates
- Counting BF can be simply adopted
  - Needs to have sufficient additional bits for counter
  - BF size is a challenge of B-NRS to be optimized

Periodic refresh for B-NRS

- Periodically rebuilds and replaces BFs using the shadow memory
- Refresh frequency can be few hours, a day, a week, a month, etc.



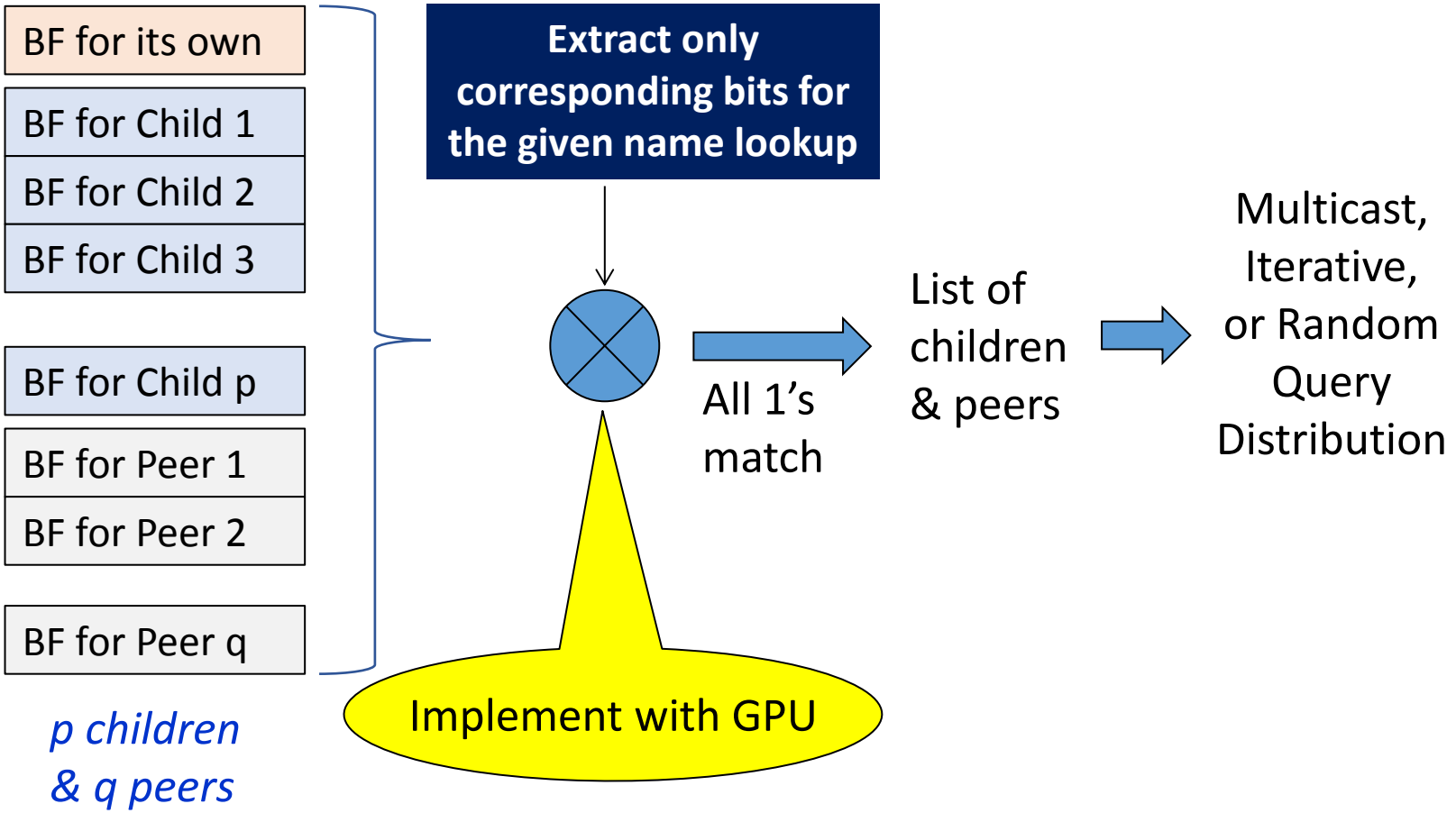
- Lossless compressed BF can be used to announce the merged BF

BF size and FPP

- *n* is the # of names in a BF: *n* = 1M
- *m* is the size of BF: *m* = 16Mb (=2MB)
- *k* is the # of hash functions: *k* = 11.09
- FPP = 0.00046 (0.046%)
- If a server on the top of the B-NRS has 1M child servers with 1M entries for each

  ➔ *1T entries* ➔ *2TB memory for FPP=0.00046*

HW assisted implementation



Questions or Comments

www.idnet.re.kr