Updated Bloom Filter based NRS for ICN

ICNRRG, Yokohama

Jungha Hong, Woojik Chun, and Heeyoung Jung

November 5, 2015
Contents

• B-NRS implementation updates since last meeting
  – BF Refresh
    • Alternative way of BF update for name deletion
  – Retransmission
    • Communication error occurs such as server shut-down, packet loss, etc.

• Discussion on NRS (or B-NRS) as a working item

• Name based web browser for ICN
  – Introduction of our implementation
B-NRS demo at ACM ICN 2015
Configuration

Server1

**GPU**: GTX TITAN X 12GB

Server2

Server3

**GPU**: GeForce 940M 2GB

Server4

Server5

Server6
## Setup

- **Name**: 24 byte flat name
- **Locator**: variable length string
- **Number of hash functions**: 11
- **False positive probability**: $\leq 4.586 \times 10^{-4}$

### Bloom Filter (BF): 2MB

| 1 | 1 | 0 | 0 | 0 | ⋮ | ⋮ | ⋮ | 1 | 1 | 0 | 1 | 0 |

### Server Setup

**BF 1**

- **BF 2**
- **BF 3000**

Server has 3000 child
Server has information for $3 \times 10^9$ names

### Lookup Table

| N1-1 | L1
| N1-2 | L2-1, L2-2
| N1-3 | L3
| ⋮   | ⋮
| N1-10^6 | L10^6 |

### BF 500

- **BF 2**
- **BF 3000**

Server has 500 child
Server has information for $5 \times 10^8$ names

### Lookup Table

| N3-1 | L1
| N3-2 | L2-1, L2-2
| N3-3 | L3
| ⋮   | ⋮
| N3-10^6 | L10^6 |
Registration

BF for Server3

BF update

BF for Server6

BF update

BF for Server4

BF update

REG (N1,L1)
Locator update

Server1

Server2

Server3

Server4

Server5

Server6

BF Check (N1) →
BF for Server4 ←find
SET (N1,L2)

FWD SET (N1,L2)

BF Check (N1) →
BF for Server6 ←find
BF Check (N1) →
BF for itself ←find

BF for itself

BF Check (N1) →

find →
find →

find →

update (N1)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>L1 : L2</td>
</tr>
</tbody>
</table>

www.idnet.re.kr
Locator lookup

Server1

Server2

Server3

Server4

Server5

Server6

BF Check (N2) →

FWD GET (N2)

FWD GET (N2)

BF Check (N2) →

No positive BF

BF Check (N2) →

FWD GET (N2)

BF Check (N2) →

No positive BF

BF Check (N2) →

FWD GET (N2)

BF Check (N2) →

No positive BF

FWD GET (N2)

BF Check (N2) →

BF for Server2

BF Check (N2) →

BF for Server5

BF Check (N2) →

No positive BF

Response (N2, L3:L4:L5)

BF for itself

No positive BF

BF Check (N2) →

BF Check (N2) →
Mobility support

BF Check (N2) → BF for Server6

BF Check (N2) → No positive BF

BF Check (N2) → BF for itself

SET (N1,L6)

FWD SET (N1,L6)

SUB (N1,L1)

BF Check (N2) → BF for itself
Deregistration

BF Check (N3) →
find
BF for Server4

BF Check (N3) →
find
No positive BF

BF Check (N3) →
find
No positive BF

BF Check (N3) →
find
No Name (N3)

BF Check (N3) →
find
No positive BF

BF Check (N3) →
find
No positive BF

BF Check (N3) →
find
No positive BF

GF for Server4

No Name (N3)

No Name (N3)

No Name (N3)

No Name (N3)
Refresh

<table>
<thead>
<tr>
<th>Server1</th>
<th>Server2</th>
<th>Server3</th>
<th>Server4</th>
<th>Server5</th>
<th>Server6</th>
</tr>
</thead>
<tbody>
<tr>
<td>FWD GET (N3)</td>
<td>No Name (N3)</td>
<td>BF Check (N3)</td>
<td>GET (N3)</td>
<td>BF Check (N3)</td>
<td>No positive BF</td>
</tr>
<tr>
<td>Refresh BF</td>
<td>Refresh BF</td>
<td>Refresh BF</td>
<td>Refresh BF</td>
<td>Refresh BF</td>
<td>Refresh BF</td>
</tr>
<tr>
<td>No positive BF</td>
<td>No Name (N3)</td>
<td>BF Check (N3)</td>
<td>No positive BF</td>
<td>No Name (N3)</td>
<td>BF Check (N3)</td>
</tr>
</tbody>
</table>
Server Disconnection

Server1

Sever2

Server3

Sever4

Server5

Server6

BF Check (N4) → find → BF for Server4

retransmission GET (N4)

Disconnection

FWD GET (N4)

Loss

N4 L8
Experimental Results with GPU (1)

<table>
<thead>
<tr>
<th>Number of BFs</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
<th>2500</th>
<th>3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total BF search time (μs)</td>
<td>40</td>
<td>42</td>
<td>38</td>
<td>39</td>
<td>43</td>
<td>41</td>
</tr>
<tr>
<td>Number of names inserted in a BF</td>
<td>$10^3$</td>
<td>$10^4$</td>
<td>$5 \times 10^4$</td>
<td>$10^5$</td>
<td>$5 \times 10^5$</td>
<td>$10^6$</td>
</tr>
<tr>
<td>A BF search time (μs)</td>
<td>45</td>
<td>46</td>
<td>44</td>
<td>47</td>
<td>45</td>
<td>46</td>
</tr>
</tbody>
</table>

CPU vs GPU

[Graph showing comparison between CPU and GPU for request time vs number of bloom filters]
Experimental Results (2)

• Scalability
  – BF search time with GPU assisted implementation is almost constant independently of the number of names and BFs

• Low latency
  – Using GPU improves the latency compared to CPU

• Locality
  – Name can be registered in any B-NRS server
Questions or Comments

www.idnet.re.kr
Further discussion

• NRS is a requirement for lookup-by-name scheme in ICN
• What about NRS (or B-NRS) as a working item in ICNRG?
• Any contributions are welcomed
Name based Web Browser for ICN

Implemented by Hae Sook Jeon
Background

• Current URL based web browser
  – Contents/web pages can not be found if
    • the server is not working properly
    • contents move to another location
    • network error such as link/port failure occurs

• ICN provides in-network caching
  – Contents/web pages can be always found as long as available in network
    • All cached copies for a content share the same name(ID)
  – *Needs a new web browser for ICN*
Name based web browser for ICN

- We implemented the following functional entities
  - Name based web browser
    - Name based web browser login/request/response
    - Name based video plugin
  - Name based web server
    - Named file registration
    - Named file provision
  - Named file caching
    - Named file local caching
    - Cached file name list management
  - Name based communication
    - GW interworking based on name
Development Environment

• OS : Ubuntu 14.04 LTS/ Android

• Developing Tools
  – QT 5.3.2 Desktop
  – QT 5.3.2 Android

• Language : C/C++

• 24 bytes flat name and B-NRS are used
Test Topology
Implementation results

• When link failure occurs,
  – File/web page was not found with HTTP
  – File/web page was found by in-network cache with IDP
Questions or Comments

www.idnet.re.kr