SAIL NetInf global connectivity – routing and forwarding

Bengt Ahlgren, SICS
(in collaboration with many SAIL-ors)
IRTF ICN RG meeting, Stockholm, Feb 15, 2013
ICN global routing scalability

• The scalability issue:
  – Sheer number of named data objects (NDOs)
    • “bookkeeping” to keep track of them
  – Cf. current Internet: number of IP addresses

• How many objects?
  – One trillion (1,000,000,000,000) unique URLs on the web
    (Google 2008)
  – At least 7 billion web pages (http://www.worldwidewebsize.com/)

• Some numbers to compare with:
  – 129 million second-level domain names in the DNS (Feb 2012)
    • Applicable if we can aggregate routing on the publisher level
  – 400K IP prefixes in the global, BGP routed, IP routing table
  – 60,000 AS numbers, of which 34,000 announced in BGP
Aggregation is key

- **For scalability**
  - Hard to handle routing state for individual NDOs
  - (Believe that NRS state for individual NDOs is possible)

- **For performance**
  - Amortise NRS cost over many individual objects
    - (note: majority of objects are small)
  - Enable fast forwarding of requests for individual objects
Notion of NDO aggregate

• A set of NDOs that for resolution and routing purposes are treated the same
  – NRS mappings and routing/forwarding information can be shared (and thus cached) for all NDOs in the aggregate

• NDO aggregates occur naturally:
  – Publishers normally make many objects available from the same origin
  – Examples: chunks of a video, photos in a collection, objects on a web page/site, and so on.

• NDOs may “belong” to more than one aggregate
NetInf routing a variant of

• GIN/REX:


• (PSIRP scopes and DONA explicit aggregation use similar idea)
Routing of NDO requests in the NetInf DFZ

- Routing/resolution in the NetInf “default-free zone”
  - corresponding to BGP-routed Internet
- Alternative to global DHT or similar solutions
  - Edge domains can use other schemes!
- Main issue: scalability
  - Need aggregation of routing information
  - Want caching in DFZ
Hybrid scheme

- Two name spaces
  - ni: naming scheme:
    - ni://example.com/sha-256;B_K97zTsFuOhug27fke4_Zgc4Myz4b_1ZNgsQjy6fkc
  - locators (IP address namespace)
- GET messages are forwarded using ni names and/or locators
  - but hard to do ni name routing in NetInf DFZ!
- Routing hint lookup service
  - global name resolution system
  - maps ni: URI authority field into a set of routing hints (locators)
Why multiple routing hints?

• explicit aggregation
  – can provide better aggregation than longest-prefix match
  – don't need full locator routing tables -> increased scalability
  – ideally use anycast and only exact match

• enable retrieving from multiple sources

• enable selecting the “best” source
  – for instance, from multiple hosting sites, or multihomed sites
• DFZ routers only need the prio=1 hints in their tables
• May want to delay adding prio 2 and 3 hints to request till actually needed
  – Means another NRS lookup at D
Where do we get next-hop from?

- **Why not directly use the routing hints?**
  - no hint forwarding table needed
  - results in sparse caching in the DFZ (one hop over DFZ)
  - less control over path taken

- **Design choice: use hint forwarding table!**
  - where the routing hints are looked up
  - require a way to populate those tables
    - or use IP forwarding table, and all IP hops will need to be NetInf routers
  - enables dense caching in the DFZ
  - enable more control over path taken
Convergence Layer Issues

• Can't assume all nodes support all CLs
  – Choice of CL is only a matter between the two nodes the CL connects
• CL is a consequence of selecting next-hop
  – *Thus can't encode CL in routing hints*
• Selection of next-hop is made using the object name, one of the routing hints, or a default entry
• Conclusion:
  – Need **next-hop table** that both selects next-hop, and which CL to use
NetInf node forwarding tables

• **Ni-name forwarding table**
  – Forwarding table for nodes using name-based forwarding

• **Locator (hint) forwarding table**
  – Forwarding table for nodes using locator-based forwarding

• NetInf nodes have *one or both* of them!
### Ni-name forwarding table

<table>
<thead>
<tr>
<th>ni-name</th>
<th>next-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>ni://example.com/sha-256;XYZ</td>
<td><a href="http://local.example.com/netinfproto/get">http://local.example.com/netinfproto/get</a></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>default</td>
<td><a href="http://gw.edge.net/netinfproto/get">http://gw.edge.net/netinfproto/get</a></td>
</tr>
</tbody>
</table>

- Exact matching
- Next-hop specifies CL and next-hop address
## Locator forwarding table

<table>
<thead>
<tr>
<th>hint</th>
<th>next-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>130.237.0.0</td>
<td><a href="http://edge.example.com/netinfproto/get">http://edge.example.com/netinfproto/get</a></td>
</tr>
<tr>
<td>10.1.10.1</td>
<td><a href="http://local.example.com/netinfproto/get">http://local.example.com/netinfproto/get</a></td>
</tr>
<tr>
<td>default</td>
<td><a href="http://gw.edge.net/netinfproto/get">http://gw.edge.net/netinfproto/get</a></td>
</tr>
</tbody>
</table>

- Exact matching
- Next-hop specifies CL and next-hop address
Forwarding process configuration

• The forwarding process is highly dependent on configuration

• What tables to use (one or both of):
  – ni-name forwarding
  – locator forwarding

• What NRS:es to consult (zero or more of):
  – any number of local ones
  – DFZ-NRS
Forwarding process

• 1. Check cache
• 2. Check ni-name forwarding table
• 3. Perform any NRS lookups
  – Resulting in additional hints
• 4. Check locator forwarding table
  – Look up all routing hints with exact match
  – Use entry that matches hint with highest priority
Scalability (admittedly handwavy)

• Number of NDO aggregates?
  – Most likely more than the number of names in DNS today
  – If using DNS – adding more leaf names should not be an issue, or?

• Number of routing hints (prio=1)
  – Network topology is not expected to change from today
  – Can therefore argue that no more needed than current number of IP prefixes
Implementation status

• Implemented as two new modules for the NEC NetInf Router Platform (NNRP)
  – hint_lookup
    • Maps the authority part of ni name to set of routing hints
    • Static table and from DNS TXT records
  – forward_lookup
    • Looks up routing hints in a forwarding table, and select the next hop
    • Static table
NetInf routing summary

• Aggregation of named data objects (NDOs)
  – Same NRS lookup and routing for all NDOs in an aggregate

• Hybrid scheme using two namespaces
  – ni: NDO naming scheme
  – locators (routing hints)

• NDO aggregate is mapped to routing hints

• Request forwarding using the hints