Routing Table Construction Method
Solely Based on **Query Flows**
for Structured Overlays

Yasuhiro Ando, Hiroya Nagao,
Takehiro Miyao, Kazuyuki Shudo

Tokyo Tech
Background: Structured Overlay

- An application-level network
  - routes a query to the responsible node.
  - enables scalable data store and messaging.
    - e.g. Distributed Hash Tables (DHT)

<table>
<thead>
<tr>
<th>Index range (digest)</th>
<th>Responsible node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab – dz</td>
<td>192.168.0.2</td>
</tr>
<tr>
<td>ea – gb</td>
<td>192.168.0.3</td>
</tr>
<tr>
<td>gc – …</td>
<td>192.168.0.4</td>
</tr>
</tbody>
</table>

“Shudo”’s tel #?

“+81 3 5734 XXXX”
Contribution

• Our finding:
  – Small number of hops does not require node distance in routing table construction and maintenance.
    • E.g. Chord based on node ID difference and Chord# based on number of nodes between nodes achieve $O(\log N)$ hops.

• Evidence:
  – Flow-based FRT (FFRT):
    A routing table construction method
  – FFRT-Chord: An FFRT-based structured overlay
Node distance based routing table construction

- Each node determines which other nodes to be on its routing table based on

- **ID difference**
  - in Chord, Kademlia, ...
  - Responsible nodes for self ID + 1, 2, 4, ..., 2^i

- **Number of nodes between the two nodes**
  - in Chord#
  - 1, 2, 4, ... 2^i nodes away from self

---

**Diagram:**
- Node distances and ID space representation.
Flow-based FRT (FFRT)

• A routing table maintenance method
  – Not node distance based
  – Instead, query flow based

• An application of Flexible Routing Tables (FRT) [P2P’11]
  – A routing table is just a list of nodes.
  – A policy given by an algorithm designer \( \leq_{ID} \) determines which nodes to be kept on a table.
  – FFRT specifies a policy based on query flows: \( \leq_{FL} \)
Query flow

- A node keeps query history: target IDs of the recent $H$ queries including queries it forwards.
- Query flow for an entry in a routing table is the number of queries in the query history to be forwarded to the entry.
Query flow based routing table maintenance

• A node refines its routing table to minimize the variance of query flows of all the entries.
  – In other words, towards a state in which all nodes on the table have equal query flows.
  – When Entry Filtering of FRT, A node selects a removed node to minimize …
  – $\leq_{FL}$: Routing tables with smaller variance precede.
Experiments

• Confirmed that an FFRT-based overlay achieves comparable hop counts with existing overlays.
  – It works well with nonuniform ID distributions because its routing maintenance is not based on IDs, though comparison with overlays for nonuniform ID distributions such as Chord# is part of future work.

• FFRT-Chord implemented on Overlay Weaver [ComCom 2008] and compared with FRT-Chord.
  – FRT-Chord shows smaller hop counts than Chord with moderate routing table sizes, 20 or larger.

Refer to the paper for graphs.
Summary and future work

• In structured overlays, small number of hops such as $O(\log N)$ does not require node distance in routing table construction and maintenance.
  – Query flow based method, FFRT and FFRT-Chord presented as an evidence.

• Future work
  – Theoretical analysis and reasoning.
    • With weighted graph, or so?