Message Bundling on Structured Overlays

Kazuyuki Shudo
Tokyo Institute of Technology

首藤 一幸
東京工業大学
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)

```
Index range (digest) | Responsible node
--- | ---
ab – dz | 192.168.0.2
ea – gb | 192.168.0.3
gc – … | 192.168.0.4
```

Routing table

```
| 1/8 |
```

“Shudo”’s tel #?

```
“+81 3 5734 XXXX”
```
Contribution: Collective Forwarding

- A **message bundling** technique for structured overlays.
  - combines multiple messages into a single message.
  - mitigates
    - the load of nodes on the **overlay network**.
    - the load of Internet routers on an **underlay network** by reducing # of packet transmission.

- Results
  - # of packet transmission: 34 % ~ 12 %
  - Data loading time: 13.0 % ~ 9.5 %
Problem: Delivery time and underlay load

• Message delivery on a structured overlay takes much time.
  – 10,000 get operations on a DHT took 40 ~ 700 sec (Section IV.C).

• An overlay imposes a burden on an underlay.
  – A message delivery requires multiple …
Proposed technique: Collective forwarding

• combines multiple messages whose next hops are the same node, and forward collectively.
  – A requesting node has a large number of requests. e.g. DB backup

# of forwarding 15 times 9 times
3 hops x 5 routes with collective forwarding

Node A
Node
Route

0 hop
1st hop
2nd hop
3rd hop

Node

F
G
H
I
J
C
D
E
B
Node A
Proposed technique: Collective forwarding

- **Bundle**
  - Messages with the same next hop.

1. Looks up next hops on routing table
2. Divides a bundle based on the next hops
3. Forwards the bundles to their next hops

<table>
<thead>
<tr>
<th>0 hop</th>
<th>Next hop is node B</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID1 ID2 ID3 ID4 ID5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st hop</th>
<th>Node C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID1 ID2 ID3 ID4 ID5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2nd hop</th>
<th>Node F G H I J</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID1 ID2 ID3 ID4 ID5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3rd hop</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ID1 ID2 ID3 ID4 ID5</td>
<td></td>
</tr>
</tbody>
</table>
Effects

• On an overlay
  – Throughput improvement
    • by handling multiple messages
    Parallel processing of multiple messages
  – Load reduction of nodes
    • by reducing message forwarding operations.
      e.g. message decode/encode, routing table lookup, ...

• On an underlay
  – Packet transmission reduction
  Load reduction
    • cf. Performance of Internet routers is shown in pps (packets per second)
Initial bundle grouping

• A bundle is continuously divided once forwarding starts.

• How does the technique compose initial bundles?
  – It is not good to combine all the millions of messages. e.g. should be < MTU with UDP

• Policy
  – Size
  – Grouping
  – When? Who?

• In our experiments
  – 10
  – Target ID-clustered and random
  – Before routing, outside an overlay
Experiments

1. **# of packet transmission** on an underlay
   IP packet delivery from a node to another node

2. **Message delivery time** on an overlay

• **Conditions**
  - 1000 nodes simulated on a single PC.
  - **Overlay Weaver** [Shudo 2008]
    • runs structured overlay routing algorithms and
    • simulates a distributed environment. E.g. comm. Latency
  - Target IDs are randomly determined.
  - Routing algorithms: Chord, Koorde, Pastry, Tapestry and Kademlia
  - Forwarding styles: iterative and recursive
• Ratio to # without the technique.

• Initial bundle grouping
  - “serial”: the technique not applied.
  - “random”
  - “clustered”: target ID-based clustering

• Consideration
  - The # was reduced to around the theoretical limit 0.1.
  - In Kademlia, a k-bucket was fulfilled and the node sends PING msg many times.

• Put and then got 50,000 data items on 1,000 nodes.

• Measured the # of packet transmission on an underlay, e.g. Internet.

Note: the forwarding style is recursive
Message delivery time

- Elapsed time to get 10,000 data items from 1,000 nodes
- 1 ms of comm. latency is simulated by Overlay Weaver.

- Two techniques for parallel processing
  - Collective forwarding
  - Multiple (10) clients, send requests in parallel

- Consideration
  - With concurrency 10, delivery speeded up 7.5 ~ 8.5 times.
  - Effects of the two techs are comparative: 7.9 sec vs. 6.9 sec.
  - Effects of the two techs are cumulative.
Related work

• **Message bundling**
  – A common technique for networks.
  – Investigated for various networks: wireless sensor network, DTN, virtual machines, ...

• **MARIF** [Mizutani 2013]
  – Bulk data transfer technique over a DHT
  – MARIF is dedicated to DHT, but collective forwarding works with structured overlays and supports multicast, for example.

• **Techniques to improve efficiency of single message delivery**
  – Proximity routing
  – 1-hop DHT
Summary

• Collective forwarding
  – combines multiple messages into a bundle and forwards it to the next hop.

• Effects
  – Improves throughput of an overlay
  – Reduces # of packet transmission on an underlay

• Experimental results
  • # of packet transmission: 34 % ~ 12 %
  • Data loading time: 13.0 % ~ 9.5 %
    – With 10 clients, 7.03 % ~ 3.12 %

Tokyo Tech