DECADE Content Replication

draft-yry-decade-replication-00

Richard Yang
Ning Zong

IETF 82th, Taipei.
Content Replication & Access

- Replication
  - To determine, which DECADE Server(s) should store a given piece of content, when, and from where does a storing DECADE Server get the content?

- Access
  - To determine, for a request for content from a Content Consumer, which DECADE Server(s) should serve the request?
We assume that DECADE should support the following recursive tree structure for content replication and access.
Design Issues on Tree Construction

- Proactive Push vs. On-demand Pull
  - Proactive Push: push content to DECADE Server(s) before Content Consumer requests.
    - Benefits: Reduce latency of content distribution;
    - Issues: May have higher wastes on storage/bandwidth if push locations are hard to predict;
  - On-demand Pull: pull content into DECADE Server(s) upon Content Consumer requests.
    - Benefits: Much lower ratio of wasted resources on storage/bandwidth; further improvement on efficiency by using Piggybacked replication;
    - Issues: May have latency issue in some real-time use cases;
A Simple Pull Example

Storage Server A

- Client P1 has account on A
- Client P2 has account on B

1. P1 puts D₁ in A

2. P2 learns that P1 has D₁

3. Request data D₁

4. Response w/ Token T₁ allowing accessing Data D₁

5. Content Request:
   - data: D₁
   - pivot account: B:P₂
   - src: Token T₁
   - dst: P₂

6. Get Object D₁ using Token T₁

7. Return Data D₁

8. Return Data D₁
General Pull Mechanisms 1/2

- DECADE-Core-Stateless Design
  - A full pull chain in DECADE content request.

<table>
<thead>
<tr>
<th>Object ID</th>
<th>NextHop S1</th>
<th>NextHop S2</th>
<th>NextHop ...</th>
</tr>
</thead>
</table>
General Pull Mechanisms 2/2

- DECADE-Core-Stateful Design
  - Introduce content forwarding table into DECADE server.
  - Control plane can update the content forwarding table.

<table>
<thead>
<tr>
<th>Object ID</th>
<th>next_hop</th>
<th>DECADE data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
Next Steps

Just a start and need further work on:

- Evaluation of controller (e.g. Application control, or DECADE Service Provider control, or hybrid mode) of replication and access tree;
- Detailed design of DECADE content forwarding table – e.g. table structure, suitable object ID naming space, etc;
- … …

Comments are highly appreciated.
Thank you