Integration Examples of In-Network Storage and P2P Applications

Presenter: Leo Chen
Objective

• A preliminary trial design to evaluate the feasibility of using in-network storage with P2P applications

• Application-agnostic storage server + application-specific control logic
  – Integration with two P2P applications
    • Vuze (a popular BitTorrent file-sharing client; plug-in style integration limits software flexibility)
    • PPNG (a P2P live streaming system based on PPLive, maintained at Yale)

• We are not pushing the protocol we used as DECADE design. It is a design exercise.
Background: In-network Storage Capabilities

- Users have accounts on the in-network storage servers
- Apps (exec’d by users) send requests to servers using token based access control
- Data object identified by hash of its content
Background: The Basic Distribution Primitive

- A basic data command primitive is to indicate a data path
Background: The Basic Distribution Primitive

- The data path primitive from Client C to server S specifies
  - `<data id>`,
  - a `<src>`,
  - an account on S `<S:account>`, and
  - a `<dst>`.
  - Interpretation
    - If `<src>` is null: it is a pure read to transfer data from `<S:account>` to `<dst>`
    - If `<dst>` is null: it is a pure write to store data from `<src>` to `<S:account>`
    - Otherwise, it is a distribution pipeline from `<src>` to `<S:account>` to `<dst>`

- Write is deduplicated

11/10/2010
Example: Endpoint Controlled Data Flow

- 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. Distribution Request:
   - data: D₁
   - pivot account: B:P2
   - src: Token T₁
   - dst: P2
6. Get Object D₁ using TokenT₁
7. Return Data D₁
8. Return Data D₁
Integration Overview
Integration Case 1: P2P Live Streaming

Network Access

Connected Peers

Token Generator
- Generate Token

Buffer

Download Scheduler

Peer Manager

Buffer Map Exchanger

Upload Scheduler

In-network Distribution Primitive

Native Transport Download
Integration Case 2: P2P File Sharing

Vuze Client
- Download Manager
- Upload Manager
- Peer Manager

Plug-in API
- Redirector
- DECADe Client

Network Access

Client
- Token Generator
  - Generate Token
- Storage Access
  - Read/Write/Distribute Objects
    - Using Distribution Primitive
Example: Data Request Flows of Vuze with Storage
## Results: P2P Live Streaming

<table>
<thead>
<tr>
<th></th>
<th>Improvement with In-network Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Startup delay</strong></td>
<td>At 80-percentile: reduced to 2/3 when no storage</td>
</tr>
<tr>
<td><strong>Piece lost rate</strong></td>
<td>About the same, at ( \leq 0.02% )</td>
</tr>
<tr>
<td><strong>Average # of freezes</strong></td>
<td>Reduced to 2/3 when no storage</td>
</tr>
</tbody>
</table>
Results: P2P File Sharing

<table>
<thead>
<tr>
<th></th>
<th>Improvement with In-network Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client upload volume</td>
<td>430 MB → 12 MB</td>
</tr>
<tr>
<td>System resource efficiency*</td>
<td>65% → 88% (35% speedup)</td>
</tr>
</tbody>
</table>

*System resource efficiency: fraction of total available upload capacity used
Thank you!