Tracker-based Peer Selection using ALTO Map Information

draft-yang-tracker-peer-selection-00

Y. Richard Yang
Richard Alimi, Ye Wang, David Zhang, Kai Lee

IETF 79 - Beijing, China - Nov 2010
Challenges

- **Tracker Scalability**
  - Many peers distributed in many torrents/channels
  - Many ISPs providing ALTO info

- **Application-Network Information Fusion**
  - Application requirements/policies
  - Application endpoint info
  - Network providers’ ALTO info
  - Third-party database info
Simple Representation

■ Peer Table

<table>
<thead>
<tr>
<th>Peer ID</th>
<th>IP Address</th>
<th>Upload Capacity</th>
<th>Play Point</th>
<th>ALTO Network Info</th>
<th>City</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>GH4C9</td>
<td>128.36.233.1</td>
<td>512 kbps</td>
<td>01:19:21</td>
<td>pid1.yale</td>
<td>New Haven</td>
<td>...</td>
</tr>
<tr>
<td>J8NRE</td>
<td>130.132.10.2</td>
<td>10 Mbps</td>
<td>00:05:37</td>
<td>pid2.yale</td>
<td>Unknown</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

■ Cost Tables, e.g.,
  ▪ ALTO Cost table

■ Problem: scanning Peer Table to select peers can be inefficient
Peer Selection using Classification

- **Objective**
  - Aggregation to improve scalability
    - only need to match categories \( \rightarrow \) **better scalability**
  - **Many Classification Attributes**, e.g.,
    - Upload capacity class, play point cluster, ALTO Network Map

- **Multi-Dimension Classification**
  - Classify peers using multiple attributes, e.g.,
    - Level 1: ALTO Network Map
    - Level 2: Upload capacity
Peer Classification: an Example

Classification 1
(upload capacity)

- n1 (Root)
  - n2 (low)
  - n3 (med)
  - n4 (high)

Classification 2
(ALTO map)

- n5 (Root)
  - n6 (ISP1)
    - n8 (PID1)
  - n7 (ISP2)
    - n9 (PID2)
    - n10 (PIDA)
    - n11 (PIDB)
Peer Classification: an Example

Classification 1 (upload capacity)
- n1 (Root)
  - n2 (low)
  - n3 (med)
  - n4 (high)

Classification 2 (ALTO map)
- n5 (Root)
  - n6 (ISP1)
    - n8 (PID1)
  - n7 (ISP2)
    - n9 (PID2)
    - n10 (PIDA)
    - n11 (PIDB)

(new peer) (high capacity) (IP address in ISP1:PID2)
Peer Selection using Classification

- **Home Node**
  - A *leaf* category node where the peer issuing LISTING request belongs to

- **Peer Selection Sequence**
  - A mapping: from a home node to a *traversal sequence* of category nodes, with a specified target fraction to be reached upon visiting each node

- **Peer Selection Process**
  - *Sequentially* follow the nodes in the sequence in order
Peer A in n4 (home leaf) requests 50 neighbors:

- select up to 25 (50%) peers from n4;
- continue to reach up to 40 (80%) from n5;
- continue to reach up to 49 (95%) from n2;
- continue to reach up to 50 (100%) from n3.

Peering Selection Table

- n4→[n4, 50%] [n5, 80%] [n2, 95%] [n3, 100%]
- n5→[n4, 20%] [n5, 60%] [n2, 95%] [n3, 100%]
- ... [n9, ...]
Simple Peer Classification using ALTO Network Maps

- One three-layer classification tree using the ALTO Network Map of each ISP
  - Used in P4P trials
  - Can be used with distributed trackers (one tracker per ISP)
Peering Matrix Computation

- **Bandwidth Matching**
  - Consider both application requirements and ALTO info
    - for each PID, tracker periodically estimates aggregated upload capacity and download demand
    - use bandwidth matching algorithm to compute weights

- **Generic Peering Matrix**
  - bandwidth matching, assuming uniform supply and demand across PIDs
ALTO/P4P Library for Tracker Peer Selection

Peer Selection:
- Upon peer LISTING request, selects peers according to classification

Peer Classification:
- Upon peer arrival, looks up new address in each ALTO Network Map

Peer Update:
- Upon peer keep alive, update statistics of classification

Application Optimization Engine
(peering matrix computation; a separate thread or machine)

ALTO Info update
(run in its own thread)
Benchmarking

1,000,000 peers; Network Map/Cost Map with 10 to 30 PIDs

Peer Selection:
- Join rate (classification + peer selection): 25,000 peers/sec in single thread

Peer Classification:
- Lookup rate: ~2,000,000 lookups/sec using Patricia tree

Memory: < 150 MB

Application Optimization Engine (if state dependent, 5 sec)

Peer Update

ALTO Info update (run in its own thread; potentially slowest)

IETF 79 - Beijing, China - Nov 2010
PlanetLab Experiments

- **Experiment Setup**
  - ~2500 P2P live streaming clients
    - 4 instances running on each PlanetLab node
  - Three emulated ALTO servers
    - US, Europe, Asia
  - Generic Peering Matrix using the library

- **Results**

<table>
<thead>
<tr>
<th>Metric</th>
<th>w/o ALTO</th>
<th>w/ ALTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Efficiency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intra-US supply ratio</td>
<td>25.9 GB (/44.1 GB)</td>
<td>40.9 GB (/52.6 GB)</td>
</tr>
<tr>
<td></td>
<td>58.7%</td>
<td>77.8%</td>
</tr>
<tr>
<td>Intra-PID supply ratio</td>
<td>6.9 GB</td>
<td>22.7 GB</td>
</tr>
<tr>
<td></td>
<td>15.6%</td>
<td>43.2%</td>
</tr>
<tr>
<td>Application Performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. Playback Startup Delay</td>
<td>31.1 seconds</td>
<td>26.9 seconds</td>
</tr>
<tr>
<td>#Playback Freezes</td>
<td>106 for all clients</td>
<td>52 for all clients</td>
</tr>
</tbody>
</table>
Thank you!
Backup Slides
Tracker ISP Data Structures

ISPPIDMap

PIDMapPortalAPI

PIDMap

ISPPDistanceMap

PDistanceMapPortalAPI

PDistanceMap

ISP 1

ISP M

IETF 79 - Beijing, China - Nov 2010
Per Channel Data Structures
## Peering Matrix

- A data structure to implement peering weights of per-ISP classification tree
  - each entry in matrix encodes peering weight from row to column
  - complexity: $O(N)$

<table>
<thead>
<tr>
<th></th>
<th>iPID1</th>
<th>iPID2</th>
<th>iPID3</th>
<th>iPID4</th>
<th>intra ISP</th>
<th>ePID1</th>
<th>ePID2</th>
<th>ePID3</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPID1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPID3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPID4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Implementation Experience

- **Processing Complexity**
  - Peer IP lookup in peer classification
    - We use Patricia Trie for IP address lookup: > 2,000,000/second
      - extended LC-trie can be more efficient
      - hash map is slow

- **Multi-thread Processing**
  - ALTO info update should run in a thread
    - periodically refresh ALTO maps
    - Network Map update triggers Cost Map update and peer classification update
      - slowest part
    - Cost Map update triggers Peering Matrix update
      - e.g., by calling AOE
  - Can run multi-thread workers for peer classification
    - if peer arrival/departure rate is high