ALTO Information Redistribution

draft-gu-alto-redistribution-01

Yingjie Gu
Richard Alimi
Roni Even
Motivation

- P2P is highly-scalable

- Used to support content distribution to millions of users
  - 2009 U.S. Presidential Inauguration
  - 2009 Spring Festival Evening in China

- Goal:
  - Enable all interested ALTO Clients to obtain desired ALTO information
ALTO Service Scalability

- Multiple methods to increase scalability are possible
  - Load balancing (deploy additional ALTO Servers)
  - Caching (use existing or deploy new web caches)

- Another technique: Utilize P2P paradigm
  - ALTO Clients may redistribute ALTO Information
Purpose of this Presentation

- Explore redistribution and identify design considerations
- Identify extensions to ALTO Protocol
- *Provide framework for discussion on redistribution*
Redistribution Framework

ISP A

ALTO Server

ALTO Client

(1) ALTO Clients mapped to ALTO Server via discovery mechanism

ISP B

ALTO Server

Redistribution Overlay

(2) An ALTO Client identifies desired ALTO information

Server : alto.isp-a.net
Info Type : Network Map
Parameters: Complete

(3) Multiple entities may be able to provide desired information
Redistribution Framework

ISP A
ALTO Server
ALTO Client

ISP B
ALTO Server

Redistribution Overlay
Redistribution Framework

(1) ALTO Clients mapped to ALTO Server via discovery mechanism
Redistribution Framework

ISP A
ALTO Server
ALTO Client

ISP B
ALTO Server

Redistribution Overlay

(2) An ALTO Client identifies desired ALTO information

Server : alto.isp-a.net
Info Type : Network Map
Parameters: Complete
Redistribution Framework

Server: alto.isp-a.net
Info Type: Network Map
Parameters: Complete

(3) Multiple entities may be able to provide desired information
Redistribution Framework

• Redistribution scheme needs some basic elements

• *ALTO Clients identify desired ALTO information*
  – Mechanism may be application-dependent

• *Transfer ALTO Information amongst ALTO Clients*
  – Mechanism may be application-dependent

• *Verify ALTO Information once received*
  – In some deployments, ALTO Clients may not be trusted
  – Suggest extensions to ALTO Protocol

• Other considerations as well (see draft for details)
ALTO Information Distribution

• ALTO Information is dependent on
  – ALTO Server (e.g., hostname and port)
  – Type (e.g., type of ALTO Query)
  – Input Parameters (e.g., ALTO Query parameters)

• **Reusable ALTO Information**
  – ALTO Information desired by multiple ALTO Clients
  – Redistributing *reusable* info can reduce ALTO Server load

• **Examples (w.r.t. draft-penno-alto-protocol-04)**
  – Reusable
    • Server Capability, Full Network Map, Full Cost Map
  – Non-reusable
    • Ranked list of endpoints (e.g., in BitTorrent)
ALTO Information Verification

- Focus on case where raw ALTO Query Response is redistributed
  - Note that P2P application *could* perform irreversible transformation (e.g., filtering) during redistribution

- ALTO Clients should be able to determine:
  - “Is this what I requested?” (server, query type, params)
  - “Did my ALTO Server generate it?” (crypto. check)

- Simple proposal (w.r.t. draft-penno-alto-protocol-04)

<table>
<thead>
<tr>
<th>Contents of ALTO Server Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestURL</td>
</tr>
<tr>
<td>Sign(PrivateKey_{ISP}, RequestURL</td>
</tr>
</tbody>
</table>

ALTO Server optionally includes additional fields

Corresponding public key retrievable by ALTO Clients
Illustrative Example: Trackerless P2P Overlay

1. client1 fetches Network Map

```
GET /prop/pid/map HTTP/1.1
```

2. client1 advertises via DHT that it has Network Map

```
put(hash("http://alto.isp-a.net/prop/pid/map"), client1.isp-a.net)
```

3. client3 queries DHT to learn where it can fetch Network Map

```
get(hash("http://alto.isp-a.net/prop/pid/map"))
```

4. client3 fetches Network Map from client1

NOTE: Other schemes such as Distributed Trackers, distribution trees, may have increased scalability
Illustrative Example:
Trackerless P2P Overlay

(1) Client1 fetches Network Map

GET /prop/pid/map HTTP/1.1
Illustrative Example:
Trackerless P2P Overlay

ALTO Server
alto.isp-a.net

P2P Client
client2.isp-a.net

DHT

P2P Client
client1.isp-a.net

P2P Client
client3.isp-a.net

(2) client1 advertises via DHT that it has Network Map
Illustrative Example: Trackerless P2P Overlay

ALTO Server
alto.isp-a.net

P2P Client
client2.isp-a.net

DHT

get(hash("http://alto.isp-a.net/prop/pid/map"))

(3) client3 queries DHT to learn where it can fetch Network Map

P2P Client
client3.isp-a.net

P2P Client
client1.isp-a.net
Illustrative Example: Trackerless P2P Overlay

NOTE: Other schemes such as Distributed Trackers or re-using existing P2P application overlay are also possible
Discussion Points

• To what extent should ALTO Protocol support redistribution?
  • Sign subsets of ALTO responses?
  • Sign only some ALTO responses?
    • E.g., for queries with limited/no input parameters
  • Sign all ALTO responses?
    • Might require specifying how to echo input parameters, etc
  • No support at all?

• Are additional requirements needed for ALTO Protocol?
Thank you