Mechanism for Peer-to-Peer Group Management using Multiple Overlays

draft-kassinen-p2prg-group-management-00

O. Kassinen, T. Koskela, E.Harjula, M. Ylianttila

Jul 28 2009, 75th IETF meeting, Stockholm
Motivation

- Major driving force for P2P: Communication and content/information sharing within and between communities

- Main motivation to form groups: to control the scope of communications for privacy reasons
  - Trust between group members

- The most common approach: group mgmt on the application layer
  - Not optimal when group members use multiple applications within the same group
  - P2P-level group management needed
Overview

- The draft introduces *a mechanism for managing peer groups in structured peer-to-peer (P2P) overlay networks*

- **Goal:** Efficient and secure interaction between the group members

- **Method:** A multiple-overlay scheme
  - *Each group forms a separate overlay* with its own address space, resources, and message routing information
  - A common overlay is used for sharing information about the group overlays

- **Advantage:** All group-specific traffic is routed inside the group overlay
  - Simplicity – no special group management protocol needed
  - Improved security and privacy
  - Routing efficiency
  - Fairer load distribution
Example Scenario

- All peers are members of the main overlay
- Peers 1 and 2 are members of group A
- Peers 2 and 4 are members of group B
- Peer 3 is not a member of any group
Group creation and maintenance

- **Group Creation** (illustrated by the figure on the right)
  - A group is created by establishing a new overlay
  - Basic case: Creator peer = bootstrap node for the group

- **Group Maintenance**
  - Publishing of group existence membership info in the main overlay
  - Normal overlay maintenance messaging in group overlay
  - One of the group peers responsible for advertisements

```
+----------------+                  +-----------------+
| Creator peer   | Lookup(group ID)   | Main overlay    |
|                | Not found          |                 |
|                | New stack instance |                 |
| Publish(group ID, group info) | Ok |                 |
| Publish("members of <group ID>", creator peer's ID) | Ok |                 |
| Publish("group list wildcard", group ID) | Ok |                 |
```
Joining and leaving

- Joining (figure)
  - Joining node finds the group by looking it up in the main overlay
  - Private group information distributed using other methods
  - Details of bootstrapping & joining depends on the protocol used in group overlay

- Leaving
  - Simply by leaving the group overlay
Group removal and binding to alias

- **Group removal**
  - Group overlay ceases to exist when no nodes remain there anymore
  - Last peer should remove group and membership information from the main overlay
    - If not, the group and membership information remains in the main overlay until their expiration

- **Binding to "Alias" ID**
  - Makes possible to establish aliases for existing groups
  - Useful in some scenarios
Data format

- XML-based documents
  - Preceded by the header "GROUP-MANAGEMENT:"
    - Enables distinguishing group management resources from other types without XML-parsing

- Example of a published group info document:

  ```xml
  GROUP-MANAGEMENT:
  <group_management type="group-info">
    <bootstrap-address>12.34.56.78</bootstrap-address>
    <bootstrap-port>5080</bootstrap-port>
    <description>The most jolly group!</description>
  </group_management>
  ```

- Detailed XML schema TBD.
Open issues

- First version of the draft => Many open issues

- E.g. Bootstrapping
  - How to ensure that only legitimate members are able to join a particular group?
  - Future work: mechanism for secure group access management
Discussion

- Questions/comments about the proposed model?

- Anybody interested in contributing to the draft?