TOC

Network Working Group	O. Kassinen
Internet-Draft	T. Koskela
Intended status: Informational	E. Harjula
Expires: January 11, 2010	M. Ylianttila
	University of Oulu
	July 10, 2009

Mechanism for Peer-to-Peer Group Management using Multiple Overlays
draft-kassinen-p2prg-group-management-00

Status of this Memo

Copyright Notice

Abstract

This document introduces a mechanism for managing peer groups in structured peer-to-peer (P2P) overlay networks. Group management enables efficient and secure interaction between the group members (peers), by controlling the scope of propagation for the group-specific

P2P messaging and other communications. The mechanism for group management, introduced in this document, is based on a multiple-overlay scheme. Each group is a separate overlay network with its own address space, resources, and message routing information. A common overlay is used for sharing information about the group overlays (sub-overlays).

Table of Contents

1. Introduction

TOC

Emerging popularity of social networking services, such as Facebook [FACEBOOK] (, “Facebook,” .), Google Groups [GOOGLEGRP] (, “Google Groups,” .), or Yahoo Groups[YAHOOGRP] (, “Yahoo Groups,” .), emphasizes the importance of group management. This is especially the case with Peer-to-Peer (P2P) networks that are natural platforms for interpersonal communications. Content and information sharing within and between communities can be seen as the major driving force behind P2P networks. Users usually have certain interests when they join the network. Thus, they inherently tend to create interest groups among each other, called communities. The users of traditional P2P applications, such as content sharing, ip telephony, or video/audio conferencing, are increasingly getting organized as peer groups.
Main motivation for users and services to form groups is to control their scope of communications for privacy reasons. Peer group members usually trust the members of the group to some extent. Groups can also

be used for security reasons, as the access to the group and thus group-specific resources and communication is usually granted only to the members of the group. Even though group management is generally considered as one of the basic functionalities of P2P services, standardization initiatives, such as IETF's P2PSIP working group [P2PSIP] (, “Peer-to-Peer Session Initiation Protocol (P2PSIP) IETF Working Group,” .), have not yet paid much attention on it.

Group management has for long been utilized in different forms in different P2P systems. The most common approach is to manage groups on application layer, i.e. using a separate protocol above P2P layer for group management. However, application-specific group management is not optimal in the situations where group members want to use multiple applications within the same group. Perhaps the best known P2P-layer group management mechanism is the one used in JXTA [JXTA] (Gong, L., “Project JXTA: A Technology Overview,” 2001.). In JXTA, each peer is a member of a global peer group, called "NetPeerGroup", and optionally one or more other peer groups. JXTA, however, does not specify in detail how to create or manage a group. With structured P2P systems, the most natural way for group management (in addition to application layer solutions) is to use separate overlays for each group.

Hierarchical overlay structures, such as [HIERAS] (Xu, Z., Min, R., and Y. Hu, “HIERAS: A DHT Based Hierarchical P2P Routing Algorithm,” 2003.) and [HierarchP2P] (Garc´es-Erice, L., Biersack, E., Felber, P., Ross, K., and L. Urvoy-Keller, “Hierarchical Peer-to-Peer Systems,” 2003.), have been proposed as a measure to improve routing efficiency of structured P2P networks. However, the fundamental model of hierarchical overlays is also useful for group management purposes.

This document describes a mechanism for managing groups of entities in structured peer-to-peer (P2P) overlay networks. In the proposed group management mechanism, each group forms a separate P2P overlay network that is specific to the group. Every member of the group participates in the routing activities and other routines of the overlay as defined by the P2P protocol and the DHT algorithm. A peer can be a member in more than one group.

A group overlay isolates the group-specific activities from the other P2P overlay based activities of the peers. This multi-overlay approach has the following advantages, compared to using a single large overlay that would contain the activities of all groups and would utilize a separate mechanism for group management.

> - Routing efficiency: Small overlays provide more efficient routing, compared to a single large overlay. Hop count is lower in small overlays.
>
> - Fairer load distribution: Only the member nodes are responsible for maintaining the specific group.
>
> - Simplicity of operation: All group-related activities are inherently contained in group-specific overlays.
>
> - Security: A base for security is provided with independent overlays that separate the communications of different groups from each other. There is no need to separately monitor the access rights to group-specific resources (except with the security measures of the selected P2P protocol that, of course, apply).
>
> - Transparency: The group management system supports an arbitrary combination of a DHT-algorithm and a DHT-based P2P overlay protocol. Furthermore, it is possible to use different DHT-algorithms within different group overlays thanks to the independent nature of the overlays.

All peers are members of the `main overlay'. The introduced mechanism uses the main overlay for distributing information about the existing groups and group memberships. Every peer has access to the publicly available information on groups, since the peers can perform searches in the main overlay.
In the following sections, certain P2P protocol related terms and action examples are used. The terms or exact details or actions might be slightly different depending on the actual P2P protocol that is used in an implementation of the group management mechanism. This document attempts to use terms and P2P actions that are common to most DHT-based P2P protocols.
As much as possible, the solution utilizes the existing DHT-based data storing (publish) and finding (lookup) mechanisms of a P2P protocol.

---

2. Terminology

TOC

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).
Overlay: A structured P2P network that is realized on top of a lower-level network (in practice an IP-supporting network).

Peer: As in P2PSIP terminology: A node that participates in the maintenance of an overlay.
Client: As in P2PSIP terminology: A node that does not participate to the maintenance of an overlay, and uses the services provided by the peers.
Group: Any group of nodes within a structured P2P network; their intercommunication is realized within a group overlay.
Group ID: A human-readable or binary identifier that identifies a group. The ID is used as a search key in the DHT system for finding information about the group overlay.
Group Description: A human-readable description of a group, bound to a group ID.
Group Overlay: An overlay that contains the member peers of a group. Group activities take place in a group overlay.
Main Overlay: A large overlay where all the peers of a certain network are members. It is used for distributing information about group overlays and group memberships.
Community: A sub-type of groups; a group that consists of human users. The terminal device of a human member acts as a peer in the group overlay.

3. Overview of the Architecture

TOC

The core idea behind the presented group management architecture is to use one DHT overlay network per community, resulting in multiple small overlays that are subsets of the main overlay. This results in an overlay hierarchy where each DHT overlay is a self-contained, independent network as illustrated in Figure 1. The group information, containing at least the contact information of the group's bootstrap peer and the group description, SHOULD be published in the main overlay. If the users, however, wish to have private groups that are not publicly advertised, the group information can remain unpublished. There can be both public and private groups in existence at the same time in a network.
The relationship of overlays is illustrated in the example in Figure 1. All peers 1, 2, 3, and 4 are members of the main overlay. Peers 1 and 2 are members in group A (and thus in group overlay A). Peers 2 and 4 are members in group B. Peer 3 is not a member in any group.

```
                +---------------------------------------------+
               /                                               \
              |                   Main Overlay                  |
               \                                               /
                +--+-----------+-------------+-------------+--+
                   |           |             |             |
             +-----+--+    +---+----+    +---+----+    +---+----+
             | Peer 1 |    | Peer 2 |    | Peer 3 |    | Peer 4 |
             +-----+--+    +---+----+    +--------+    +---+----+
                   |           |     \                     |
               +---+-----------+      +--------------------+
              /                 \    /                      \
             |  Group Overlay A  |  |     Group Overlay B    |
              \                 /    \                      /
               +---------------+      +--------------------+
```

Example situation: peers as members in different overlays.

Figure 1

4. Overview of the Operations

TOC

4.1. Creation of Groups

TOC

A new group is created by establishing a new overlay network (the group overlay), where the group activities will take place. Such creation usually implies that a new instance of the P2P protocol stack is created so that its routing tables, resource tables, etc. are specific to the created group. Before creating a new group, the creator checks with a resource lookup request from the main overlay that no other group already exists with the same ID.

In the basic case, the creator peer will act as the bootstrap node for the group. This means that the node will respond to the joining nodes' initial requests when they try to become members of the group. The creator peer does not need to be the bootstrap peer; the role can be assumed by any other peer of the group.

The creator peer publishes a DHT key-value pair of <group ID, group info> to the main overlay. Group info contains the bootstrap node's ID and its contact information i.e. the IP address and port. In addition, the creator peer publishes the members list, which contains only the first member, itself, at this point. The list is published under a DHT key that contains an equivalent of the phrase "members of" along with the group ID; thus it is possible for others to search for the members of a specified group. Finally, the creator peer publishes a key-value pair where the key is a wildcard that enables listing all groups and the value is the group ID. `Wildcard' here means that the same key is used for all groups created by any peer; thus, searching for the key will yield a list of resources, each related to a specific group. The action sequence of creating a public group is illustrated in Figure 2.

If a node that cannot be responsible for overlay management (for example, a low-capability P2PSIP client) wants to create a group, it needs to ask another node (with more capabilities or public IP address) to become the creator of the group. This more capable node will, at least temporarily, take the responsibility of maintaining the group information in the main overlay. A node can also be a client in the main overlay and a peer in a group overlay, or vice versa.

Groups can be public or private. Private groups are not published to the main overlay. The same logic applies to memberships. Even if a group is public, the memberships might be private, thus the members list is not published in that case.

```
Creator Peer                                           Main Overlay
     |                                                       |
     | Lookup(group ID)                                      |
     |------------------------------------------------------>|
     | 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                                                    |
     |<------------------------------------------------------|
     |                                                       |
```

Creation of a public group.

Figure 2

4.2. Maintenance of Groups

TOC

Maintenance of a group refers to the publishing of group existence and group membership information in the main overlay. In addition, group maintenance also comprises the overlay maintenance messaging (e.g. keepalive signals between neighbor nodes) as required by the used P2P protocol. Publishing is typically done repeatedly, if the overlay's P2P protocol requires resource objects to be refreshed periodically.

In every group there MUST be at least one peer that is capable of maintaining the group overlay's routing and resource-management functionalities. If all peers leave the group overlay so that only clients remain, at least one of the clients SHOULD become a peer, or alternatively, peer services MAY be requested from an external peer (this functionality is not defined in this version of this draft). If neither of the mentioned conditions is not fulfilled, the group ceases to exist.

If the advertiser peer leaves the group overlay and this group is a public group, another peer SHOULD assume the role of advertiser peer; otherwise, the refreshing of the public group info in the main overlay ceases.

4.3. Joining and Leaving from Groups

TOC

Joining is typically done via the bootstrap node of the group. This node is initially the creator peer. Joining can occur through any member peer of the group (if the underlying P2P protocol allows this), but usually the joining peer knows specifically about the bootstrap peer that is advertised in the main overlay. In the case of public groups, information about the existing nodes (members) in the group is found in the main overlay. In the case of private groups, the information can be conveyed in some other way, for example, by an e-mail or a paper note that instructs the joiner to contact a specific existing overlay-node. The action sequence of joining a public group is shown in Figure 3.

Leaving a group is done by the departing from the corresponding group overlay, using the leave function of the underlying P2P protocol. A leaving node MUST also stop refreshing its group membership publication in the main overlay, and SHOULD, if possible, remove the information's existing version from the main overlay. If the leaving node participated as a peer in the group overlay, it also naturally retires from the maintenance functionalities, which happens when it leaves the overlay.

```
   Joining Peer                       Main Ov.  Boots. Peer  Group Ov.
        |                                |           |           |
        | Lookup(group ID)               |           |           |
        |------------------------------->|           |           |
        | Group info                     |           |           |
        |<-------------------------------|           |           |
        |                                |           |           |
        | New stack instance             |           |           |
        |------+                         |           |           |
        |<-----+                         |           |           |
        |                                |           |           |
        | Bootstrap request              |           |           |
        |------------------------------------------->|           |
        | OK                             |           |           |
        |<-------------------------------------------|           |
        |                                |           |           |
        | Join overlay                   |           |           |
        |------------------------------------------------------->|
        | OK                             |           |           |
        |<-------------------------------------------------------|
        |                                |           |           |
        | Publish("members...", peer ID) |           |           |
        |------------------------------->|           |           |
        | OK                             |           |           |
        |<-------------------------------|           |           |
        |                                |           |           |
```

Joining a public group.

Figure 3

Pointers [I-D.hardie-p2psip-p2p-pointers] (Hardie, T. and V. Narayanan, “Pointers for Peer-to-Peer Overlay Networks, Nodes, or Resources,” March 2009.) in P2P networks have been proposed as a mechanism for referring to a specific overlay network or resource in a way such that these pointers can be used in communication outside the P2P network. In group management, pointers can be used for referring to a specific group. For example, a pointer-based "link" containing an overlay pointer could be used when inviting new members to a group overlay. Opening the "link" then results in the instantiation of the DHT-utilizing protocol stack (and/or other similar required actions) and performing the overlay joining operation of the used P2P protocol.

4.4. Removal of Groups

TOC

Removing a group is done by shutting down the group overlay so that no nodes remain there anymore. Group and membership information SHOULD be actively removed from the main overlay. This means removing all of the following: 1) the resource (key-value pair) whose key is the group ID in the main overlay; 2) the group from the list contained in the resource whose key is the groups wildcard; and 3) the membership resources whose key is "members of <group ID>".

The group overlay can also cease to exist even without an explicit removal of the group. If no peers remain in the overlay, i.e. only clients remain and no nodes willing to act as new peers are found, the group disappears.

4.5. Binding of Groups to an "Alias" ID

TOC

It is possible to establish aliases for existing groups with a `bind' action. Binding is done by publishing to the main overlay a key-value pair, where the key is the new group ID (bind ID, i.e. an alias of the existing overlay) and the value contains the original overlay ID along with a tag which indicates that the key-value pair is a result of binding. This is similar to the key-value pair published during group creation, but instead of group info, the value only contains the above-mentioned information. Thus, when a peer joins to a group by identifying the group with a bind-ID, the peer learns the original (actual) ID of the group overlay and must initiate another Lookup request to find the group info of the group overlay.

Binding is useful when, for example, creating location-based communities where the ID of a location-specific Bluetooth beacon device or an RFID label is used for triggering joining to an existing group overlay; the bind ID in this case would be the Bluetooth ID or the ID of the RFID label.

5. Data Formats

TOC

In order to enable the peers to interpret the group-related pieces of information correctly, the related DHT-stored values (in the key-value pairs that are mentioned in the various group operations in this document) have some syntax requirements that are specified in this section.

XML is used for encoding the group-related information in the resource values. All these XML documents are preceded by the header "GROUP-MANAGEMENT:" (outside of the XML document), which enables a peer to

recognize group-management related resources from other kinds of resources, simply by checking if the beginning of the resource value matches "GROUP-MANAGEMENT:". This also eliminates the need for trying XML parsing in the case of non-group-related resources. In the future, a dedicated MIME type might be registered, to be used instead of the "GROUP-MANAGEMENT:" header that currently serves as a type identifier. Inside all these XML documents, the root element is called "group_management". The root element has an attribute called "type" that indicates, how to interpret the inner elements. In group info (published during group creation), the attribute value is "group_info". In group binding, the attribute value is "group_id_redirect". This way, when joining a group, the joining peer can determine if the found XML resource contains directly the contact information of a group (the group ID was used as the lookup key), or does it contain the actual ID of a bound group, which of course requires a second lookup request in order to find the group info of the group in question (the bind ID was used as the lookup key).
The following XML document is an example of a published group info.

```
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>
```

The following XML document is an example of a published binding. When a lookup is performed with the bind ID as the search key, this resource value is found. The value of the attribute "encoding" can be either "as-is", indicating that the contained group ID (possibly intended as human-readable) can be used as such for the lookup of actual group info, or "hex", indicating that the contained group ID (possibly binary) is encoded in hexadecimal characters, two hex characters per byte.

```
GROUP-MANAGEMENT:
<group_management type="group-id-redirect">
  <actual-group-id encoding="hex">0276EC1F</actual-group-id>
</group_management>
```

The case of joining to a group with an ID that is a result of a binding is not shown in the figures of this document. Figure 3 only showed the case, where the group info is directly found.

Thanks to the overlay-per-group scheme, the need for special data formats is minimal, since the group-internal activities (resource publishing for communications, collaboration, etc.) are free from syntax requirements from the viewpoint of group management. This means that those resources do not need any special tagging to indicate "this data belongs to the group Foo", because all group-specific acitivity is automatically contained in the group overlay and is not risked to being confused with other overlays' activities.

6. Performance Considerations

TOC

Optimization techniques for facilitating a node's concurrent membership in multiple overlays can be used. For example, to decrease memory consumption, routing information about remote nodes can be shared between the locally stored data structures in such a way that maintaining duplicate pieces of information about remote nodes is avoided. However, this kind of techniques are optional. A straightforward implementation can simply utilize several concurrently running P2P protocol stacks that are not coordinated among each other. In a simple case where a peer is a member in the main overlay and one group overlay, such an implementation would run two protocol stack instances concurrently.

7. Acknowledgements

TOC

The authors wish to acknowledge the valuable comments of Prof. Henning Schulzrinne, Columbia University, NY, USA.

8. IANA Considerations

TOC

This memo includes no request to IANA at this time.

9. Security Considerations

TOC

The introduced group management mechanism effectively encapsulates group-specific traffic inside a group overlay, and thus helps to improve the security and privacy. However, the problem of bootstrapping, i.e. how to ensure that only legitimate members are able to join a particular group, remains. The future work includes developing mechanisms for secure group access management.

10. References

TOC

10.1. Normative References

TOC

[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).

10.2. Informative References

TOC

[FACEBOOK]	“Facebook.”
[GOOGLEGRP]	“Google Groups.”
[HIERAS]	Xu, Z., Min, R., and Y. Hu, “HIERAS: A DHT Based Hierarchical P2P Routing Algorithm,” 2003.
[HierarchP2P]	Garc´es-Erice, L., Biersack, E., Felber, P., Ross, K., and L. Urvoy-Keller, “Hierarchical Peer-to-Peer Systems,” 2003.
[I-D.hardie-p2psip-p2p-pointers]	Hardie, T. and V. Narayanan, “Pointers for Peer-to-Peer Overlay Networks, Nodes, or Resources,” internet-draft hardie-p2psip-p2p-pointers-01, March 2009.
[JXTA]	Gong, L., “Project JXTA: A Technology Overview,” 2001.
[P2PSIP]	“Peer-to-Peer Session Initiation Protocol (P2PSIP) IETF Working Group.”
[YAHOOGRP]	“Yahoo Groups.”

Authors' Addresses

TOC

	Otso Kassinen
	University of Oulu
	Erkki Koiso-Kanttilan katu 3
	University of Oulu, 90014
	Finland
Phone:	+358 8 553 2811
Email:	otso.kassinen@ee.oulu.fi

	Timo Koskela
	University of Oulu
	Erkki Koiso-Kanttilan katu 3
	University of Oulu, 90014
	Finland
Phone:	+358 8 553 2522
Email:	timo.koskela@ee.oulu.fi
	Erkki Harjula
	University of Oulu
	Erkki Koiso-Kanttilan katu 3
	University of Oulu, 90014
	Finland
Phone:	+358 8 553 2522
Email:	erkki.harjula@ee.oulu.fi
	Mika Ylianttila
	University of Oulu
	Erkki Koiso-Kanttilan katu 3
	University of Oulu, 90014
	Finland
Phone:	+358 8 553 25311
Email:	mika.ylianttila@ee.oulu.fi