PPSP
INTERNET-DRAFT
Intended Status: Standards Track
Expires: January 16, 2014

Rachel Huang
Ning Zong
Huawei
Rui S. Cruz
Mario S. Nunes
IST/INESC-ID/INOV
Joao P. Taveira
July 15, 2013

PPSP Tracker Protocol--Extended Protocol
draft-huang-ppsp-extended-tracker-protocol-04

Abstract

This document specifies an extended Peer-to-Peer Streaming Protocol - Tracker Protocol, which is a new extension protocol complementing the basic core messages and usages specified in the base tracker protocol for the exchange of meta information between trackers and peers, such as initial offer/request of participation in multimedia content streaming, content information, peer lists and reports of activity and status. It extends the base tracker protocol to include new optional messages providing new usages in the communications between peer and tracker. The extension protocol is retro-compatible with the base tracker protocol.

Table of Contents

1. Introduction

The PPSP Tracker Protocol is one of the Peer-to-Peer Streaming Protocol which specifies standard format/encoding of information and messages between PPSP peers and PPSP trackers. Based on the requirements defined in [I-D.ietf-ppsp-problem-statement], the base tracker protocol specified in [I-D.ietf-ppsp-base-tracker-protocol] has provided the basic core messages to be exchanged between trackers and peers in order to carry out some fundamental operations. They're mandatory messages covering most basic and universal use cases and MUST be implemented in all PPSP-based streaming systems.

This document specifies some extensions to complement the basic core messages and usages specified in [I-D.ietf-ppsp-base-tracker-protocol]. Some new optional messages are extended to provide new usages in some dedicated scenarios. The extension protocol is retro-compatible with the base tracker protocol. Messages using this specification MUST be safely rejected by trackers which don't support this specification without affecting interoperability.

2. Terminology

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 RFC 2119 [RFC2119].

This draft uses terms defined in [I-D.ietf-ppsp-problem-statement] and [I-D.ietf-ppsp-base-tracker-protocol].

3. Motivation

There are a number of possible usages and issues which may be useful for discussion and which the base tracker protocol may not be able to deal with.

1. The peer list of a specific swarm obtained by a peer may be out of date. It requires the peer asks the tracker for a updated one. For example, a peer is streaming some content. After a while, it finds out that it couldn't connect to some other peers any more because they're stopping sharing the content. Losing most connections with remote peers will lead to service quality decline. In this case, it is required for the peer to have some mechanisms to update the peer list.

2. A peer may have the requirement to inform the tracker its new network address when the peer has changed its primary network attachment. One example is that a peer with a LAN and a WiFi

interface which are going through different routers. The peer is using some PPSP-based P2P application which can keep working when the peer switches from the LAN to the WiFi (for example, unplugging the Ethernet cable, the P2P connection can recover automatically).

3. In the base tracker protocol, the disconnection between peer and tracker is achieved by the timeout of STAT_REPORT messages, which means that trackers lack the ability to free resources timely. In some cases when the number of connected peers has reached the maximum capability of a tracker, resources of the tracker could not be released immediately even if some peers have already left the connections. Some P2P applications may require to overcome the shortage of the base tracker protocol.

4. A peer may have the requirement to stream the content from some specific point. For example, an end user previously watched a content and stopped watching it unfinishedly (having disconnected). But the next day he would expect to continue watching it from where he interrupted. So the peer may prefer the tracker could select proper peers for specific content in a swarm.

The above use cases require the base tracker protocol to be extended.

4. The extended Tracker Protocol Overview

4.1. Extended Request Messages

4.1.1. Enhanced Request Messages

In this section, the request messages specified in the base tracker protocol are extended to meet the needs of use cases listed in section 3.

1. CONNECT

This message tends to solve the issue 4 raised in section 3. The extension is that CONNECT Request message now include the information of specific content scopes, either media content representations or specific chunks of a media representation in a swarm. The format and detailed processing of CONNECT Request message will be further discussed in Section 5.1.

2. STAT_REPORT

The STAT_REPORT Request message is extended to allow the exchanges of content data information, like chunkmaps, between an active peer and a tracker. The information can be used by a tracker as a qualification to select appropriate peer lists when peers request to

the tracker for the peer lists of some specific contents. An example of a STAT_REPORT for multiple properties is illustrated in subsection 4.5.

4.1.2. New Request Messages

Two new messages, are introduced in this section to extend those specified in the base tracker protocol [I-D.ietf-ppsp-base-tracker-protocol].

1. FIND

The FIND Request message allows a peer to request the tracker for the peer list of a swarm when it has already joined the swarm. The request can include specific content scopes, either media content representations or specific chunks of a media representation in a swarm, and may also include the new network address of the peer. On receiving a FIND message, the tracker finds the peers, listed in content status of the specified swarm that can satisfy the requesting peer's requirements, returning the list to the requesting peer. To create the peer list, the tracker may take peer status, capabilities and peers priority into consideration. Peer priority may be determined by network topology preference, operator policy preference, etc.

2. DISCONNECT

The DISCONNECT Request message is used when the peer intends to no longer participate in all swarms. When receiving the message from a peer, the tracker deletes the corresponding activity records related to the peer (including its status and all content status for the corresponding swarms). In such a case, DISCONNECT message will have the same effect of timer expiring (STAT_REPORT), but providing a graceful disconnect from the system.

4.2. Usage of Extended Request Messages

An example of usage of the extended request messages is the illustrated in Figure 1.

In that figure a peers starts by connecting to the system and joining a specific swarm (swarm_a) in SEED mode.

While active the peer periodically updates the tracker using STAT_REPORT messages. Later, the peer CONNECTs another swarm (swarm_b) but in LEECH mode, i.e., the end-user intended to watch that content while still sharing the first one. During the stream the peer requests an updated list of peers in that swarm to the

tracker.

When the peer wants to leave the second content unfinished, the peer sends CONNECT message with leave the corresponding swarm (swarm_b) action while still sharing the first content (swarm_a).

Later the peer DISCONNECTs from the system.

Next time when the peer wants to continue watching the content it previously streamed, the peer CONNECT the corresponding swarm in LEECH mode with the interrupted chunk information.

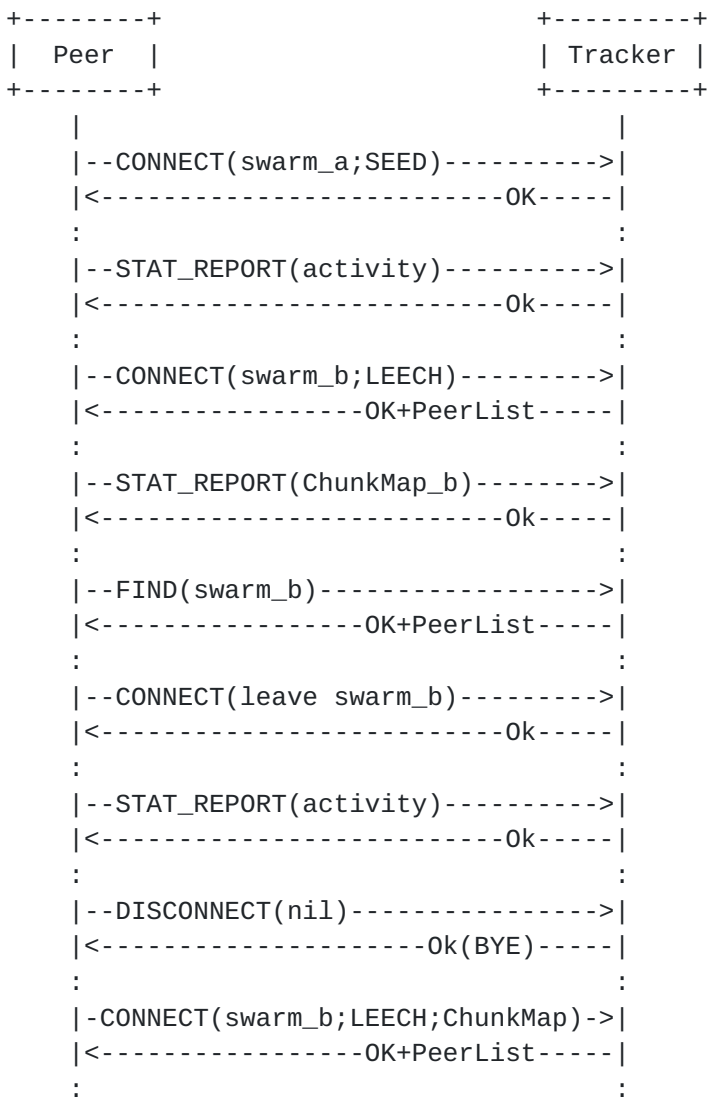

Figure 1: Example of a session for a extended PPSP-TP.

4.3. Extended Tracker Transaction State Machine

The tracker state machine has been introduced in the base tracker protocol [I-D.ietf-ppsp-base-tracker-protocol]. Every tracker MUST keep a tracker state machine in which the state transitions are triggered by peer registrations. In addition to the tracker state machine, a transaction state machine for each registered Peer-ID is also specified. In this specification, as some additional messages have been introduced and some basic messages have been extended, an updated "per-Peer-ID" transaction state machine (Figure 2) is specified to provide more functionality and detailed control to the tracker protocol. This extended "per-Peer-ID" transaction state machine is compatible with the ones specified in the base tracker protocol.

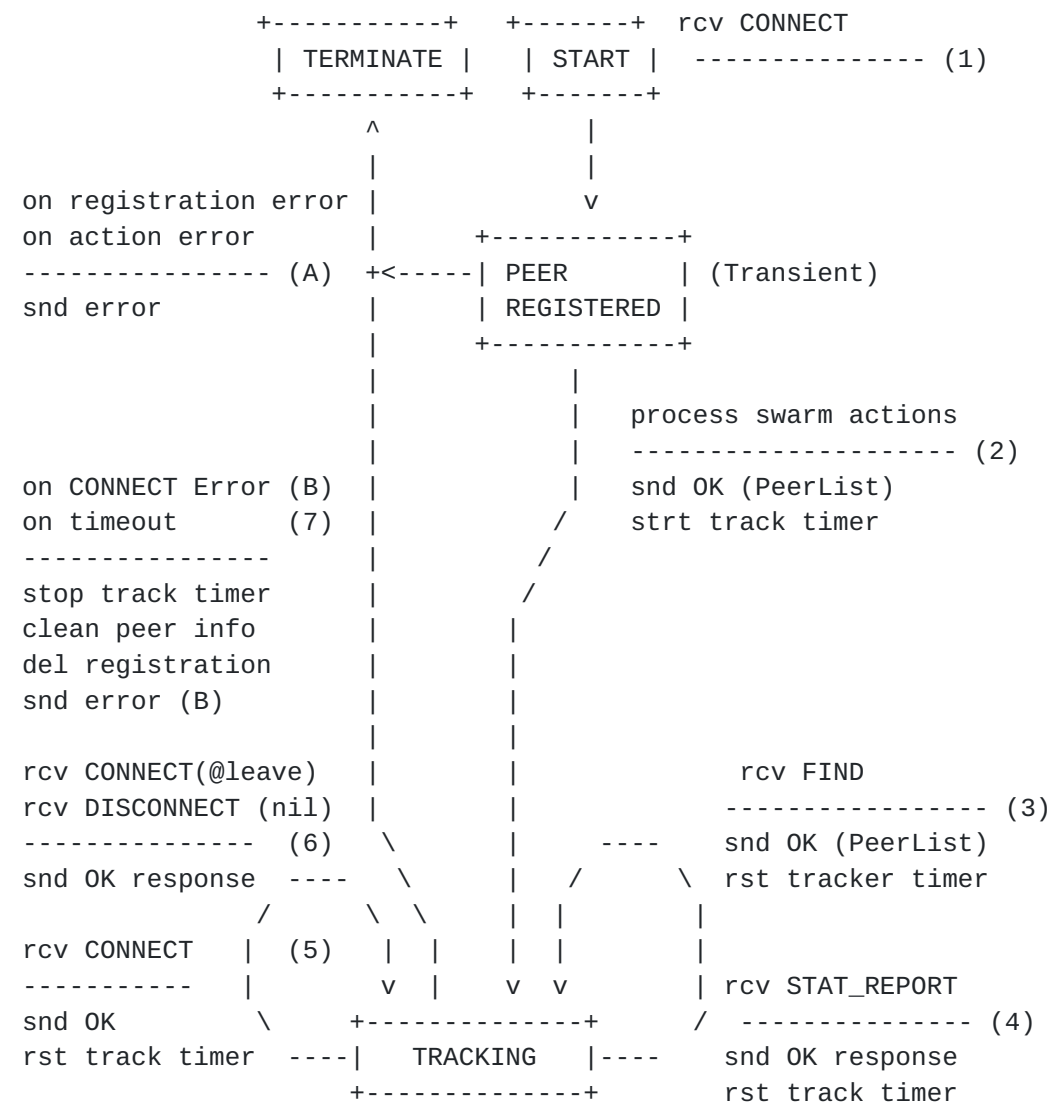

Figure 2: Extended Per-Peer-ID Transaction State Machine

The state diagram in Figure 2 illustrates the complete state changes together with the causing events and resulting actions when implementing basic tracker protocol with extended protocol. Note that Specific error conditions are not shown in the state diagram.

4.3.1. Normal Operation

On normal operation the extended process consists of the following steps:

1) This step is same with step 1) in section 6.1 of the base tracker protocol [I-D.ietf-ppsp-base-tracker-protocol].

2) This step is same with step 2) in section 6.1 of the base tracker protocol [I-D.ietf-ppsp-base-tracker-protocol].

3) While TRACKING, a FIND message received with valid swarm information from the peer resets the "track timer" and is responded with a successful condition, for including the appropriate list of peers for the scope in the FIND request.

4) While TRACKING, a STAT_REPORT message received from the peer resets the "track timer" and is responded with a successful condition. The STAT_REPORT message MAY contain information related with Swarm-IDs to which the peer is joined.

5) While TRACKING, a CONNECT message received with valid swarm actions information from the peer resets the "track timer" and is responded with a successful condition.

6) While TRACKING, a DISCONNECT message received from the peer, or a CONNECT message with the leave the last swarm action, the tracker stops the "track timer", cleans the information associated with the participation of the Peer-ID in the the swarm(s) joined, responds with a successful condition, deletes the registration of the Peer-ID and transitions to TERMINATED state for that Peer-ID.

7) In TRACKING state, without receiving STAT_REPORT messages from the peer, on timeout (track timer) the tracker cleans all the information associated with the Peer-ID in all swarms it was joined, deletes the registration, and transitions to TERMINATE state for that Peer-ID.

4.3.2. Error Conditions

A) At the PEER REGISTERED state (while no response has been sent) receiving FIND, CONNECT, STAT_REPORT messages from the peer is considered as an error condition. The tracker responds with error code 403 Forbidden. Also when a CONNECT Request only contains invalid swarm actions, the tracker will do the same things.

B) At the TRACKING state (while the "track timer" has not expired) receiving an CONNECT message form the peer with invalid swarm actions (e.g., joining multiple swarms as LEECH) or receiving an FIND message from the peer with invalid Swarm_Id which hasn't been joined in is considered an error condition. The tracker responds with error code 403 Forbidden, stops the "tracker timer", deletes the registration and transitions to TERMINATE state for that Peer-ID.

NOTE: This situation may correspond to a malfunction at the peer

or to malicious conditions. A preventive measure would be to reset the "track timer" one last time and if no valid message is received proceed to TERMINATE state for the Peer-ID by de-registering the peer and cleaning all peer information.

4.4. Request/Response Syntax and Format

The architecture specified in the base tracker protocol [I-D.ietf-ppsp-base-tracker-protocol] doesn't have to be extended. It still uses the same two-layer architecture of the base tracker protocol. Besides that, the message syntax is mainly identical with that used by the base tracker protocol except some elements are extended to contain the new optional and updated messages:

The SwarmID element MUST be present in FIND requests, but may be present in DISCONNECT requests.

The PeerNum element MAY be present in FIND requests and MAY contain the attribute @abilityNAT to inform the tracker on the preferred type of peers, in what concerns their NAT traversal situation, to be returned in a peer list.

The PeerGroup element MUST be present in CONNECT requests and responses, and MAY be present in FIND requests and MAY be present in responses to FIND requests if the corresponding response returns information about peers.

One element "ContentGroup" is added to the format of Request. It MAY be present in requests referencing content, i.e., CONNECT and FIND, if the request includes a content scope.

The extended semantics of the attributes and elements within a PPSPTrackerProtocol root element is described in subsection 4.3.1.

4.4.1. Extended Semantics of PPSPTrackerProtocol Elements

Some of the semantics defined in the the base tracker protocol MUST be extended. The extension semantics of PPSPTrackerProtocol elements are described in bellow.

Element Name or Attribute Name	Use	Description
PPSPTrackerProtocol	1	The root element.
@version	M	Provides the version of PPSP-TP.

Request	0...1	Provides the request method and MUST be present in Request.
Response	0...1	Provides the response method and MUST be present in Response.
TransactionID	M	Root transaction Identification.
Result	0...N	Result of @action MUST be present in Responses.
@transactionID	CM	Identifier of the @action.
PeerID	0...1	Peer Identification. MUST be present in Request.
SwarmID	0...N	Swarm Identification. MUST be present in Requests.
@action	CM	Must be set to JOIN or LEAVE.
@peerMode	CM	Mode of Peer participation in the swarm, "LEECH" or "SEED".
@transactionID	CM	Identifier for the @action.
PeerNUM	0...1	Maximum peers to be received with capabilities indicated.
@abilityNAT	CM	Type of NAT traversal peers, as "No-NAT","STUN","TURN" or "PROXY"
@concurrentLinks	CM	Concurrent connectivity level of peers, "HIGH", "LOW" or "NORMAL"
@onlineTime	CM	Availability or online duration of peers, "HIGH" or "NORMAL"
@uploadBWlevel	CM	Upload bandwidth capability of peers, "HIGH" or "NORMAL"
PeerGroup	0...1	Information on peers (Table 3)
ContentGroup	0...1	Information on content (Table 4)
StatisticsGroup	0...1	Statistic data (Table 5)

Legend:
Use for attributes: M=Mandatory, OP=Optional,
CM=Conditionally Mandatory
Use for elements: minOccurs...maxOccurs (N=unbounded)
Elements are represented by their name (case-sensitive)
Attribute names (case-sensitive) are preceded with an @

Table 1: Semantics of the Extended PPSPTrackerProtocol.

The semantics of PeerGroup element is almost identical with that in the base tracker protocol. It is listed below for convenience of reading.

Element Name or Attribute Name	Use	Description

PeerGroup	0...1	Contains description of peers.
PeerInfo	1...N	Provides information on a peer.
@swarmID	0...1	Swarm Identification.
PeerID	0...1	Peer Identification. MAY be present in responses.
PeerAddress	0...N	IP Address information.
@addrType	M	Type of IP address, which can be "ipv4" or "ipv6"
@priority	CM	The priority of this interface. Used for NAT traversal.
@type	CM	Describes the address for NAT traversal, which can be "HOST" "REFLEXIVE" or "PROXY".
@connection	OP	Access type ("3G", "ADSL", etc.)
@asn	OP	Autonomous System number.
@ip	M	IP address value.
@port	M	IP service port value.
@peerProtocol	OP	PPSP Peer Protocol supported.

Legend:
Use for attributes: M=Mandatory, OP=Optional, CM=Conditionally Mandatory
Use for elements: minOccurs...maxOccurs (N=unbounded)
Elements are represented by their name (case-sensitive)
Attribute names (case-sensitive) are preceded with an @

Table 2: Semantics of PeerGroup.

Table 3 describes the semantics of StatisticsGroup element. StatisticsGroup element has been extended to contain content information which indicate by "Representation" attribute.

Element Name or Attribute Name	Use	Description
StatisticsGroup	0...1	Provides statistic data on peer and content.
Stat	1...N	Groups statistics property data.
@property	M	The property to be reported property values and elements in Table 5 of [I-D.ietf-ppsp-base-tracker-protocol]
Representation	0...N	Describes a component of content.
@id	CM	Unique identifier for this Representation.
SegmentInfo	1...N	Provides segment information by

			segment range. The chunkmap can be encoded in Base64 [RFC4648].
	@startIndex	CM	The index of the first media segment in the chunkmap report for this Representation.
	@endIndex	CM	The index of the last media segment in the chunkmap report for this Representation.
	@chunkmapSize	CM	Size of chunkmap reported.

Legend:
Use for attributes: M=Mandatory, OP=Optional, CM=Conditionally Mandatory
Use for elements: minOccurs...maxOccurs (N=unbounded)
Elements are represented by their name (case-sensitive)
Attribute names (case-sensitive) are preceded with an @

Table 3: Semantics of StatisticsGroup.

ContentGroup is a new element extended in this specification. The semantics of this element is described in Table 4.

Element Name or Attribute Name	Use	Description
ContentGroup	0...1	Provides information on content.
Representation	1...N	Describes a component of content.
@id	M	Unique identifier for this Representation.
SegmentInfo	1	Provides segment information.
@startIndex	M	The index of the first media segment in the request scope for this Representation.
@endIndex	OP	The index of the last media segment in the request scope for this Representation.

Legend:
Use for attributes: M=Mandatory, OP=Optional, CM=Conditionally Mandatory
Use for elements: minOccurs...maxOccurs (N=unbounded)
Elements are represented by their name (case-sensitive)
Attribute names (case-sensitive) are preceded with an @

Table 4: Semantics of ContentGroup

The Representation element describes a component of a content identified by its attribute @id in the MPD. This element MAY be present for each component desired in the scope of the FIND request. The scope of each Representation is indicated in the SegmentInfo element by the attribute @startIndex and, optionally, @endIndex.

The peer may use this information in CONNECT or FIND requests, for example, to join a swarm starting from a specific point and to find adequate peers in the swarm for that content scope.

4.4.2. Extended Request/Response Element in Request Messages

Table 5 specifies the valid string representations for the requests extended in this specification to complement those define in the base tracker protocol. These values MUST be treated as case-sensitive.

Extended XML Request Methods String Values
DISCONNECT
FIND

Table 5: Extended Valid Strings for Request Element of Requests.

The response elements in response messages are identical with those specified in the base tracker protocol, which can be found in subsection 7.2.3 of [I-D.ietf-ppsp-base-tracker-protocol].

4.5. Compatibility with the Base Tracker Protocol

Trackers are RECOMMNEDED to implement extended tracker protocol to be compatible with either peers using base tracker protocol or peers using extended tracker protocol. But it is not mandatory. Peers implementing the extended tracker protocol sending enhanced request messages and new request messages to legacy trackers will get respond with 400 (Bad request, with reason-phrase "Unknown Messages"), which indicate the messages couldn't be recognized by the tracker. In this case, the peers MUST stop interacting with the track in extended request messages while using base tracker protocol to do communications.

5. Request/Response Processing

5.1. Enhanced CONNECT Request

This method is used when a peer wants to join one or multiple swarms. The tracker records the Peer-ID, connect-time, IP addresses and link status.

The peer MUST properly form the XML message-body, set the Request method to CONNECT, generate and set the TransactionID, and set the PeerID with the identifier of the peer. The peer SHOULD also include the IP addresses of its network interfaces in the CONNECT message.

Extended CONNECT request is retro-compatible with the CONNECT request message defined in the base tracker protocol specification.

An example of the message-body of the extended CONNECT Request is the following.

```
<?xml version="1.0" encoding="UTF-8"?>
<PPSPTrackerProtocol xmlns="TBD"
                     schemaLocation="TBD"
                     version="1.0">
  <Request>CONNECT</Request>
  <PeerID>656164657220</PeerID>
  <TransactionID>12345.0</TransactionID>
  <SwarmID action="JOIN" peerMode="LEECH"
           transactionID="12345.1">1111</SwarmID>
  <PeerNum abilityNAT="STUN"
           concurrentLinks="HIGH"
           onlineTime="NORMAL"
           uploadBWlevel="NORMAL">5</PeerNum>
  <ContentGroup>
    <Representation id="tag0">
      <SegmentInfo startIndex="20" />
    </Representation>
    <Representation id="tag6">
      <SegmentInfo startIndex="20" />
    </Representation>
  </ContentGroup>
  <PeerGroup>
    <PeerInfo>
      <PeerAddress addrType="ipv4" ip="192.0.2.1" port="80"
                   priority="1" />
      <PeerAddress addrType="ipv6" ip="2001:db8::1" port="80"
                   priority="2"
                   type="HOST"
                   connection="ADSL" />
    </PeerInfo>
  </PeerGroup>
</PPSPTrackerProtocol>
```

In this example, the peer wants to participate in swarm 1111 to watch the program as LEECH, and it also wishes to start from a specific point of the content. So the CONNECT request message contains a ContentGroup element to include the information which could be used by the tracker to restrict the searching for peer list. The extended CONNECT request MAY include a PeerNum element to indicate to the tracker the number of peers to be returned in a list corresponding to the indicated properties, being @abilityNAT for NAT traversal (considering that PPSP-ICE NAT traversal techniques may be used), and optionally @concurrentLinks, @onlineTime and @uploadBWlevel for the preferred capabilities. For the case that PeerMode is LEECH, the tracker will search and select a proper list of peers satisfying the conditions requested. The peer list MUST contain the Peer-IDs and the corresponding IP addresses. To create the peer list, the tracker may take peer status and network location information into consideration, to express network topology preference or operators' policy preferences, with regard to the possibility of connecting with other IETF effors such as ALTO [I.D.ietf-alto-protocol]. Thus a PeerGroup MAY also be needed in an extended CONNECT request messages.

The response MUST have the same TransactionID value as the request. This specification doesn't do any extension to the response of CONNECT messages. An example of a Response message for the extended CONNECT Request from a leecher is:

```
<?xml version="1.0" encoding="UTF-8"?>
<PPSPTrackerProtocol xmlns="TBD"
                     schemaLocation="TBD"
                     version="1.0">
  <Response>SUCCESSFUL</Response>
  <TransactionID>12345</TransactionID>
  <PeerGroup>
    <PeerInfo>
      <PeerID>656164657220</PeerID>
      <PeerAddress addrType="ipv4" ip="198.51.100.1" port="80"
                   priority="1"
                   type="REFLEXIVE"
                   connection="3G"
                   asn="64496" />
    </PeerInfo>
    <PeerInfo>
      <PeerID>956264622298</PeerID>
      <PeerAddress addrType="ipv4" ip="198.51.100.22" port="80"
                   asn="64496" />
    </PeerInfo>
    <PeerInfo>
```

```
        <PeerID>3332001256741</PeerID>
        <PeerAddress addrType="ipv4" ip="198.51.100.201" port="80"
                     asn="64496" />
      </PeerInfo>
    </PeerGroup>
  </PPSPTrackerProtocol>
```

5.2. DISCONNECT Request

This method is used when the peer intends to leave the system and no longer participate.

The tracker SHOULD delete the corresponding activity records related with the peer in the corresponding swarms (including its status and all content status).

The peer MUST properly form the XML message-body, set the Request method to DISCONNECT, set the PeerID with the identifier of the peer, randomly generate and set the TransactionID.

An example of the message-body of a DISCONNECT Request for the peer leaving all joined swarms is the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<PPSPTrackerProtocol xmlns="TBD"
                     schemaLocation="TBD"
                     version="1.0">
  <Request>DISCONNECT</Request>
  <PeerID>656164657221</PeerID>
  <TransactionID>12345</TransactionID>
</PPSPTrackerProtocol>
```

An example of a Response message for the DISCONNECT Request is the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<PPSPTrackerProtocol xmlns="TBD"
                     schemaLocation="TBD"
                     version="1.0">
  <Response>SUCCESSFUL</Response>
  <TransactionID>12345</TransactionID>
<PPSPTrackerProtocol>
```

5.3. FIND Request

This method allows peers to request to the tracker, whenever needed, a new peer list for the swarm or for specific scope of chunks of a

media content representation of that swarm.

The peer MUST properly form the XML message-body, set the request method to FIND, set the PeerID with the identifier of the peer, set the SwarmID with the identifier of the swarm the peer is interested in. And optionally, in order to find peer having the specific chunks, the peer may include the ContentGroup element in the JOIN request message to indicate a specific point in the stream.

This message is mainly used for leechers to update the peer list. It is unnecessary to set the PeerMode element in FIND request messages.

The peer MUST generate and set the TransactionID for the request.

An example of the message-body of a FIND Request is the following:

```
    <?xml version="1.0" encoding="UTF-8"?>
    <PPSPTrackerProtocol xmlns="TBD"
                         schemaLocation="TBD"
                         version="1.0">
      <Request>FIND</Request>
      <PeerID>656164657221</PeerID>
      <SwarmID>1111</SwarmID>
      <TransactionID>12345</TransactionID>
      <PeerNum abilityNAT="STUN"
               concurrentLinks="HIGH"
               onlineTime="NORMAL"
               uploadBWlevel="NORMAL">5</PeerNum>
      <ContentGroup>
        <Representation id="tag4">
          <SegmentInfo startIndex="110" endIndex="150" />
        </Representation>
      </ContentGroup>
    </PPSPTrackerProtocol>
```

The FIND request MAY include a PeerNum element to indicate to the tracker the number of peers to be returned in a list corresponding to the indicated properties, being @abilityNAT for NAT traversal (considering that PPSP-ICE NAT traversal techniques may be used), and optionally @concurrentLinks, @onlineTime and @uploadBWlevel for the preferred capabilities.

In the case of a FIND with a specific scope of a stream content the request SHOULD include a ContentGroup to specify the content Representations segment range of interest.

When receiving a well-formed FIND Request the tracker processes the information to check if it is valid. In case of success a response

message with a Response value of SUCCESSFUL will be generated and the tracker will include the appropriate list of peers satisfying the conditions requested. The peer list returned MUST contain the Peer-IDs and the corresponding IP Addresses.

The tracker may take peer status and network location information into consideration when selecting the peer list to return, to express network topology preferences or Operators' policy preferences, with regard to the possibility of connecting with other IETF efforts such as ALTO [I.D.ietf-alto-protocol].

An example of a Response message for the FIND Request is the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<PPSPTrackerProtocol xmlns="TBD"
                     schemaLocation="TBD"
                     version="1.0">
  <Response>SUCCESSFUL</Response>
  <TransactionID>12345</TransactionID>
  <PeerGroup>
    <PeerInfo>
      <PeerID>956264622298</PeerID>
      <PeerAddress addrType="ipv4" ip="198.51.100.22" port="80"
                   asn="64496" />
    </PeerInfo>
    <PeerInfo>
      <PeerID>3332001256741</PeerID>
      <PeerAddress addrType="ipv4" ip="198.51.100.201" port="80"
                   asn="64496" />
    </PeerInfo>
  </PeerGroup>
</PPSPTrackerProtocol>
```

The Response MUST include a PeerGroup with PeerInfo data that includes the public IP address of the selected active peers in the swarm.

The tracker MAY also include the attribute @asn with network location information of the transport addresses of the peers, corresponding to the Autonomous System Numbers of the access network provider of each peer in the list.

The response MAY also include a PeerGroup with PeerInfo data that includes the requesting peer public IP address. If STUN-like function is enabled in the tracker, the PeerAddress includes the attribute @type with a value of REFLEXIVE, corresponding to the transport address "candidate" of the peer.

An example of a Response message for the FIND Request including the requesting peer public IP address is the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<PPSPTrackerProtocol xmlns="TBD"
                     schemaLocation="TBD"
                     version="1.0">
  <Response>SUCCESSFUL</Response>
  <TransactionID>12345</TransactionID>
  <PeerGroup>
    <PeerInfo>
      <PeerID>656164657221</PeerID>
      <PeerAddress addrType="ipv4" ip="198.51.100.1" port="80"
                   priority="1"
                   type="REFLEXIVE"
                   connection="3G"
                   asn="64496" />
    </PeerInfo>
    <PeerInfo>
      <PeerID>956264622298</PeerID>
      <PeerAddress addrType="ipv4" ip="198.51.100.22" port="80"
                   asn="64496" />
    </PeerInfo>
    <PeerInfo>
      <PeerID>3332001256741</PeerID>
      <PeerAddress addrType="ipv4" ip="198.51.100.201" port="80"
                   asn="64496" />
    </PeerInfo>
  </PeerGroup>
</PPSPTrackerProtocol>
```

5.4. Enhanced STAT_REPORT Request

This message still uses the specifications of the base tracker protocol [I-D.ietf-ppsp-base-tracker-protocol]. The Stat element has been extended with one property, "ContentMap", to allow peers reporting map of chunks they have. The tracker would not have the ability to treat the FIND requests for specific content chunks, unless peers report this kind of information. Examples are provided below.

An example of the message-body of an enhanced STAT_REPORT request is the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<PPSPTrackerProtocol xmlns="TBD"
                     schemaLocation="TBD"
                     version="1.0">
```

```
<Request>STAT_REPORT</Request>
<PeerID>656164657221</PeerID>
<TransactionID>12345</TransactionID>
<StatisticsGroup>
  <Stat property="StreamStatistics">
    <SwarmID>1111</SwarmID>
    <UploadedBytes>512</UploadedBytes>
    <DownloadedBytes>768</DownloadedBytes>
    <AvailBandwidth>1024000</AvailBandwidth>
  </Stat>
  <Stat property="StreamStatistics">
    <SwarmID>2222</SwarmID>
    <UploadedBytes>1024</UploadedBytes>
    <DownloadedBytes>2048</DownloadedBytes>
    <AvailBandwidth>512000</AvailBandwidth>
  </Stat>
  <Stat property="ContentMap">
    <SwarmID>1111</SwarmID>
    <Representation id="tag0">
      <SegmentInfo startIndex="0" endIndex="24"
                   chunkmapSize="25">
        A/8D/wP/A/8D/wP/A/8D/wP/A/8D/wP/....
      </SegmentInfo>
    </Representation>
    <Representation id="tag1">
      <SegmentInfo startIndex="0" endIndex="14"
                   chunkmapSize="15">
        A/8D/wP/A/8D/wP/A/8D/wP/A/8D/wP/....
      </SegmentInfo>
      <SegmentInfo startIndex="20" endIndex="24"
                        chunkmapSize="5">
        A/8D/wP/A/8D/wP/A/8D/wP/A/8D/wP/....
      </SegmentInfo>
    </Representation>
  </Stat>

    <Stat property="ContentMap">
      <SwarmID>2222</SwarmID>
      <Representation id="tag5">
      <SegmentInfo startIndex="0" endIndex="4"
                   chunkmapSize="5">
        A/8D/wP/A/8D/wP/A/8D/wP/A/8D/wP/....
      </SegmentInfo>
    </Representation>
    <Representation id="tag6">
      <SegmentInfo startIndex="0" endIndex="4"
                   chunkmapSize="5">
        A/8D/wP/A/8D/wP/A/8D/wP/A/8D/wP/....
```

```
        </SegmentInfo>
      </Representation>
    </Stat>
  </StatisticsGroup>
</PPSPTrackerProtocol>
```

If the request is valid the tracker process the received information for future use, and generates a response message with a Response value of SUCCESSFUL.

The response MUST have the same TransactionID value as the request.

An example of a Response message for the START_REPORT Request is the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<PPSPTrackerProtocol xmlns="TBD"
                     schemaLocation="TBD"
                     version="1.0">
  <Response>SUCCESSFUL</Response>
  <TransactionID>12345</TransactionID>
</PPSPTrackerProtocol>
```

4.6. Error and Recovery Conditions

This document does not introduce any new error and recovery conditions. The implementation of error treatment MUST refer to the base tracker protocol specification [I-D.ietf-ppsp-base-tracker-protocol], subsection 8.6.

5. Security Considerations

The extended tracker protocol proposed in this document introduces no new security considerations beyond those described in the base tracker protocol specification [I-D.ietf-ppsp-base-tracker-protocol].

6. IANA Considerations

There are presently no IANA considerations with this document.

7. Acknowledgments

The authors would like to thank many people for their help and comments, particularly: Zhang Yunfei, Martin Stiemerling, Johan Pouwelse and Arno Bakker.

The authors would also like to thank the people participating in the EU FP7 project SARACEN (contract no. ICT-248474) [refs.saracenwebpage] for contributions and feedback to this document.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the SARACEN project or the European Commission.

8 References

8.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[ISO.8601.2004] International Organization for Standardization, "Data elements and interchange formats - Information interchange - Representation of dates and times", ISO Standard 8601, December 2004.

8.2 Informative References

[I-D.ietf-ppsp-problem-statement] Zhang, Y., Zong, N., Camarillo, G., Seng, J., and Y. Yang, "Problem Statement of P2P Streaming Protocol (PPSP)", draft-ietf-ppsp-problem-statement-13 (work in progress), February 2013.

[I-D.ietf-ppsp-base-tracker-protocol] Cruz, R., Nunes, M., Gu, Y., Xia, J., and J. Taveira, "PPSP Tracker Protocol-Base Protocol (PPSP-TP/1.0)", draft-ietf-ppsp-base-tracker-protocol-00 (work in progress), February 2013.

[I.D.ietf-alto-protocol] Alimi, R., Penno, R., Yang, Y., "ALTO Protocol", draft-ietf-alto-protocol-11, (work in progress), March 2012.

[ISO.IEC.23009-1] ISO/IEC, "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats", ISO/IEC DIS 23009-1, Aug. 2011.

[refs.saracenwebpage] "SARACEN Project Website", http://www.saracen-p2p.eu/.

Authors' Addresses

Rachel Huang
Huawei
Phone: +86-25-56623633
EMail: rachel.huang@huawei.com

Rui Santos Cruz
IST/INESC-ID/INOV
Phone: +351.939060939
Email: rui.cruz@ieee.org

Ning Zong
Huawei
Phone: +86-25-56624760
EMail: zongning@huawei.com

Mario Serafim Nunes
IST/INESC-ID/INOV
Rua Alves Redol, n.9
1000-029 LISBOA, Portugal
Phone: +351.213100256
Email: mario.nunes@inov.pt

Joao P. Taveira
IST/INOV
Email: joao.silva@inov.pt