

PPSP  
INTERNET-DRAFT  
Intended Status: Standards Track  
Expires: September 10, 2015

Rachel Huang  
Huawei  
Rui Cruz  
Mario S. Nunes  
INESC-ID/INOV  
Joao P. Taveira  
IST/INOV  
Lingli Deng  
China Mobile  
March 9, 2015

**PPSP Tracker Protocol-Extended Protocol  
draft-huang-ppsp-extended-tracker-protocol-08**

**Abstract**

This document specifies an extension to the PPSP Tracker Protocol - Base Protocol, which complements the core messages of the protocol with Request-Response enhancements and usages, and with a new DISCONNECT Protocol-level message. These enhancements and usages are related with the exchange of meta information between trackers and peers, such as initial offer/request of participation in multimedia content streaming, content information, peer lists, reports of activity and status, and graceful disconnection from the network. The extension is retro-compatible with the PPSP-TP Base Protocol.

**Status of this Memo**

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/1id-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at

<http://www.ietf.org/shadow.html>

#### Copyright and License Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.



## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Terminology . . . . .</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Motivation . . . . .</a>	<a href="#">4</a>
<a href="#">4.</a>	<a href="#">Extended Tracker Protocol Overview . . . . .</a>	<a href="#">5</a>
<a href="#">4.1.</a>	<a href="#">Request-Response Extension . . . . .</a>	<a href="#">5</a>
<a href="#">4.2.</a>	<a href="#">Protocol-level Extension . . . . .</a>	<a href="#">6</a>
<a href="#">4.3.</a>	<a href="#">Usage of Extended Request Messages . . . . .</a>	<a href="#">6</a>
<a href="#">4.4.</a>	<a href="#">Extended Tracker Transaction State Machine . . . . .</a>	<a href="#">7</a>
<a href="#">4.4.1.</a>	<a href="#">Normal Operation . . . . .</a>	<a href="#">9</a>
<a href="#">4.4.2.</a>	<a href="#">Error Conditions . . . . .</a>	<a href="#">10</a>
<a href="#">5</a>	<a href="#">Extended Tracker Protocol Specification . . . . .</a>	<a href="#">10</a>
<a href="#">5.1.</a>	<a href="#">Request/Response Syntax and Format . . . . .</a>	<a href="#">10</a>
<a href="#">5.3.</a>	<a href="#">Compatibility with the Base Tracker Protocol . . . . .</a>	<a href="#">12</a>
<a href="#">5.4.</a>	<a href="#">Negotiation of Chunk Addressing Methods . . . . .</a>	<a href="#">13</a>
<a href="#">5.5.</a>	<a href="#">Request/Response Processing . . . . .</a>	<a href="#">14</a>
<a href="#">5.5.1.</a>	<a href="#">Enhanced FIND Request . . . . .</a>	<a href="#">14</a>
<a href="#">5.5.2.</a>	<a href="#">Enhanced STAT_REPORT Request . . . . .</a>	<a href="#">14</a>
<a href="#">5.5.3.</a>	<a href="#">DISCONNECT Request . . . . .</a>	<a href="#">14</a>
<a href="#">6.</a>	<a href="#">Error and Recovery Conditions . . . . .</a>	<a href="#">15</a>
<a href="#">7.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">15</a>
<a href="#">8.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">15</a>
<a href="#">9.</a>	<a href="#">Acknowledgments . . . . .</a>	<a href="#">15</a>
<a href="#">10</a>	<a href="#">References . . . . .</a>	<a href="#">15</a>
<a href="#">10.1</a>	<a href="#">Normative References . . . . .</a>	<a href="#">15</a>
<a href="#">10.2</a>	<a href="#">Informative References . . . . .</a>	<a href="#">16</a>
	<a href="#">Authors' Addresses . . . . .</a>	<a href="#">16</a>



## **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 [RFC 6972](#) [[RFC6972](#)], 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. The core messages are mandatory, covering most basic and universal use cases, and MUST be implemented in all PPSP-based streaming systems.

This document specifies extensions to the base core messages of [[I-D.ietf-ppsp-base-tracker-protocol](#)] with enhancements in request/responses and new optional request message, providing new usages in some scenarios. The extension of the base protocol is retro-compatible with the PPSP-TP Base Protocol, and messages using this specification MUST be safely rejected by trackers not supporting the extensions to avoid 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 [[RFC6972](#)] 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. In the base tracker protocol, the disconnection between peer and tracker is achieved by a timeout (of periodic STAT\_REPORT messages) which means that trackers lack the ability to timely free up resources. In some cases when the number of connected peers is reaching the maximum capacity of a tracker, resources of the tracker cannot be released immediately even if some peers leave the swarm, hence resulting in connection failures. This requires the base tracker protocol to be extended to have a message providing the ability to notify the tracker that a peer has left.
2. A peer may have the requirement to start streaming the content from one specific point of the content timeline. For example, when the end-user watches only part of a content and decides to



stop and leave, or pauses for a long time. When the end-user wants to resume the session he/she expects to continue watching the content from the point where he/she interrupted. The peer may then request the tracker to select a subset of peers capable to provide that specific content scope. In this case, it requires that the quest for content from neighbor peers should contain the content scope information and peers should constantly report their content scope information to the tracker.

#### **4. Extended Tracker Protocol Overview**

The extended Tracker Protocol consists of two Request-Response Extensions (to the FIND and STAT\_REPORT Request messages of the Base Protocol) and one Protocol-level Extension (a new DISCONNECT Request message).

##### **4.1. Request-Response Extension**

In this section, the FIND and STAT\_REPORT messages specified in the base tracker protocol are extended to meet the needs of use case 2 listed in [section 3](#).

**FIND:** The enhanced FIND Request message allows a peer to request the tracker for a subset of peers in a swarm but including specific content scopes, either media content representations or specific chunks/segments of a media representation in a swarm, and may also include an updated network address of the peer. On receiving a FIND message, the tracker selects a subset of peers satisfying the requesting scope. To create the peer list, the tracker may also take peer status, capabilities and peers priority into consideration. Peer priority may be determined by network topology preference, operator policy preference, etc. The format and detailed processing of enhanced CONNECT Request message is presented in [Section 5.2](#).

**STAT\_REPORT:** The enhanced STAT\_REPORT Request message allows 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 subsets of peers in the swarm satisfying specific scopes (in terms of content). The format and detailed processing of enhanced CONNECT Request message is presented in [Section 5.3](#).

To present the content data information, the chunk addressing schemes (section 4 of [[I-D.ietf-ppsp-peer-protocol](#)]) are used to support different ways of identifying chunks and expressing chunk availability of a peer in a compact fashion. The chunk addressing





methods for certain content should be recorded in the metadata of the swarm for the content, and they can be obtained by peers or trackers during the enrollment and bootstrap stage.

#### **4.2. Protocol-level Extension**

A new Request message is introduced in this section to extend those specified in the base tracker protocol [I-D.ietf-ppsp-base-tracker-protocol], to meet the needs of issue 1 listed in [section 3](#).

**DISCONNECT:** The DISCONNECT Request message is used when the peer intends to no longer participate in all swarms. When receiving the DISCONNECT Request 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, the DISCONNECT Request message will have the same effect of timer expiring (STAT\_REPORT), but providing a graceful disconnection of that peer from the system.

#### **4.3. Usage of Extended Request Messages**

An example of usage of the extended request messages is illustrated in Figure 1. In that figure a peer starts by connecting to the system and joining a specific swarm (swarm\_a) in SEED mode (step\_1).

While active, the peer periodically updates the tracker using STAT\_REPORT messages. Later, the peer CONNECTs to another swarm (swarm\_b) but in LEECH mode, i.e., the end-user intends to watch that new content while still sharing the first one (step\_2). During the streaming session the peer requests an updated list of peers in that new swarm to the tracker (step\_3). When the end-user wants to leave the second content, not having even finished watching, the peer sends a CONNECT message with a "leave" action (step\_4) for the corresponding swarm (swarm\_b) but remains sharing the first content (swarm\_a).

But later, the end user wants to continue watching the content he/she previously left unfinished. So the peer firstly CONNECTs to the corresponding swarm in LEECH mode, then sends FIND message with the specific content information scope (step\_5). Tracker will find the group of peers who has the specific content for this peer.

Finally, the peer wants to quit the system completely. It does not have to send a CONNECT message with a "leave" action one by one. It just sends the DISCONNECT message to the tracker (step\_6), then tracker will remove all the information for this peer.



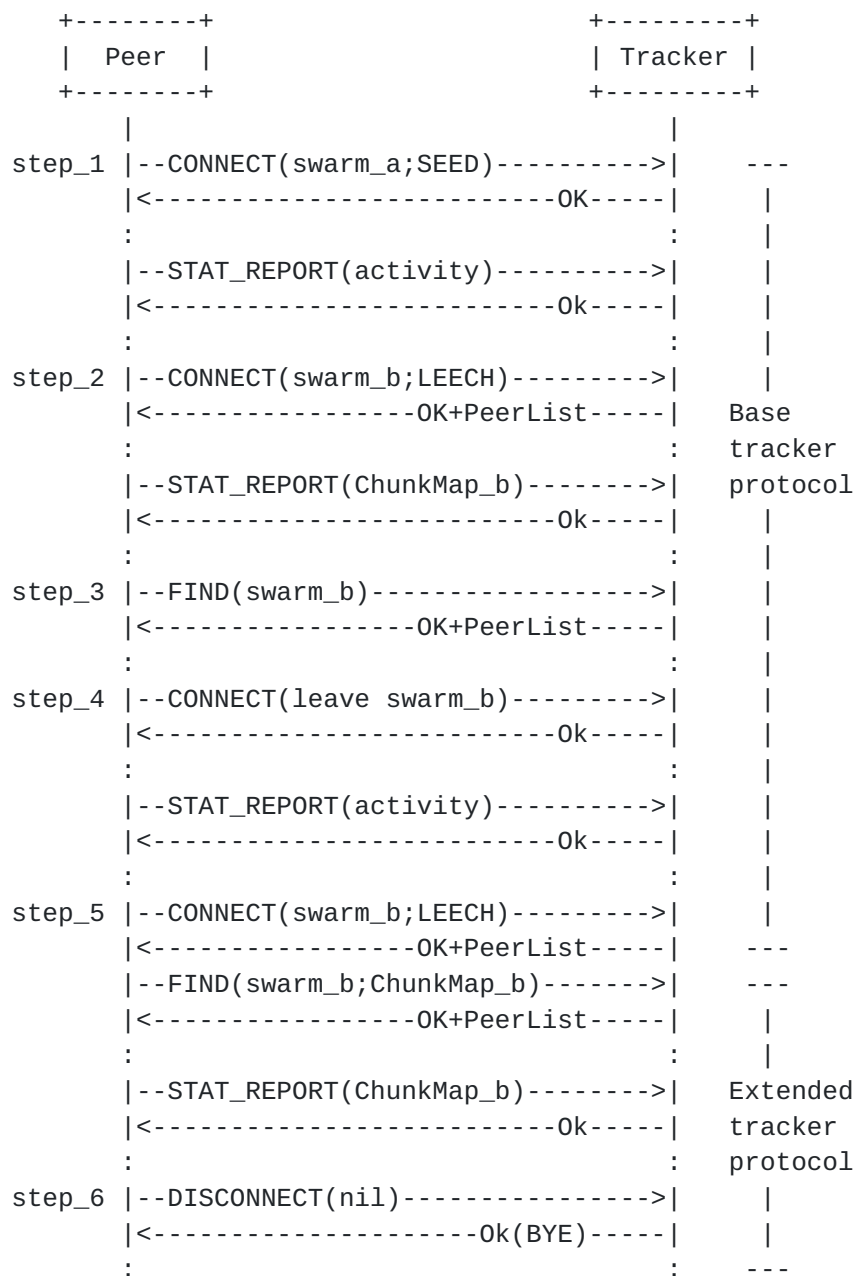


Figure 1: Example of a session for a extended PPSP-TP.

#### 4.4. Extended Tracker Transaction State Machine

The tracker state machine introduced in the base tracker protocol [[I-D.ietf-ppsp-base-tracker-protocol](#)] is now updated in this specification to reflect the extensions introduced. An updated "per-Peer-ID" transaction state machine (Figure 2) is described, corresponding to the enhanced functionalities and control steps of the extended tracker protocol. This extended "per-Peer-ID"



transaction state machine is compatible with the one specified in the base tracker protocol.

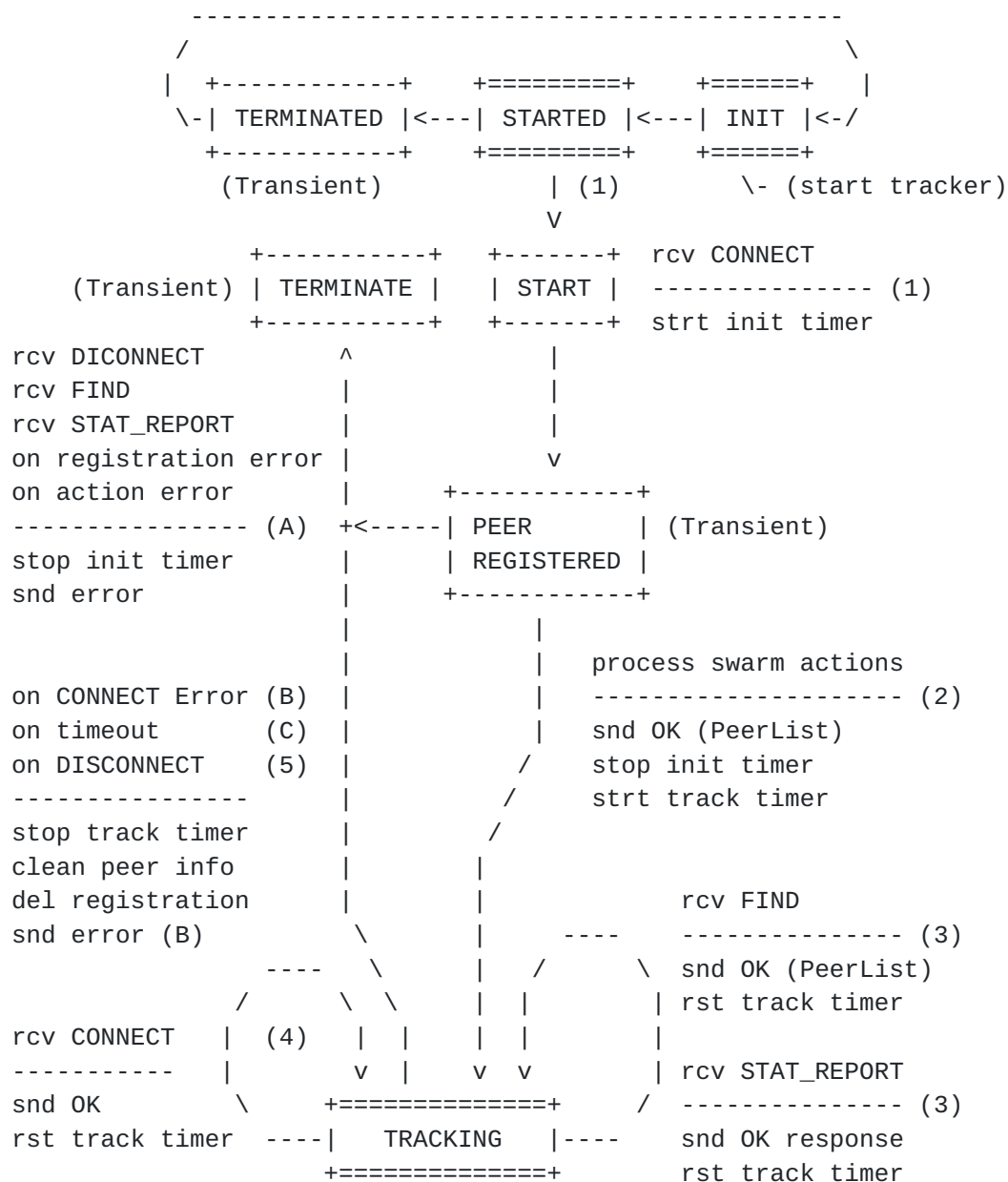


Figure 2: Extended "Per-Peer-ID" Transaction State Machine

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

#### 4.4.1. Normal Operation

Normal operation steps are the same with [section 2.3.1](#) of the base tracker protocol [[I-D.ietf-ppsp-base-tracker-protocol](#)] except adding





step 5 as follow:

- 5) While TRACKING, a DISCONNECT message received from the peer, or a CONNECT message with the action to leave the last swarm, 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.

#### **4.4.2. Error Conditions**

Error condition steps are the same with [section 2.3.2](#) of the base tracker protocol [[I-D.ietf-ppsp-base-tracker-protocol](#)] except the 2nd paragraph of step A:

- A) At PEER REGISTERED state, if the Peer ID is considered invalid (in the case of a CONNECT request, or FIND request, or STAT\_REPORT request, or DISCONNECT request received from an unregistered Peer ID), the tracker responds with either error codes authentication required or Forbidden (described in [section 4.3](#) of the base tracker protocol), transitions to TERMINATE state for that Peer ID and that state machine instance is destroyed.

### **5 Extended Tracker Protocol Specification**

#### **5.1. Request/Response Syntax and Format**

The architecture specified in the base tracker protocol [[I-D.ietf-ppsp-base-tracker-protocol](#)] does not suffers any modification in the extended protocol. The syntax is identical with some elements extended to contain new optional attributes:

The request type includes CONNECT, FIND, STAT\_REPORT and DISCONNECT, including a "content" element to the FIND method, that MAY be present in requests referencing content, i.e., FIND and STAT\_REPORT, if the request includes a content scope.

The extended semantics of the request therefore is described as follows.

```
ppsp_tp_string_t ppsp_tp_request_type_t =    "CONNECT"
                                              | "FIND"
                                              | "STAT_REPORT"
                                              | "DISCONNECT";
```

```
Object {
    ppsp_tp_version_t      version;
    ppsp_tp_request_type_t request_type;
```



```

        ppsp_tp_string_t      transaction_id;
        ppsp_tp_string_t      peer_id;
        [JSONValue request_data = ppsp_tp_req_connect  connect
                                | ppsp_tp_req_find    find
                                | ppsp_tp_stat_group_t stat_report;]
    } ppsp_tp_request;

```

Note that only when request\_type of ppsp\_tp\_request is "DISCONNECT", request\_data is allowed to be omitted.

The extended value for ppsp\_tp\_version\_t is listed in Table 1 as follow:

+-----+	
ppsp_tp_version_t	Description
+-----+	
0	Reserved
1	PPSP base tracker protocol
2	PPSP extended tracker protocol
	(specified in this document )
3-255	Unassigned
+-----+	

Table 1: Extended PPSP Tracker Protocol Version Numbers

The semantics for the content information element is described as follow:

```

Object {
    ppsp_tp_integer_t  start_index;
    ppsp_tp_integer_t  end_index; // 0 means no end
}ppsp_tp_segment_info_t;

Object {
    ppsp_tp_integer_t      chunk_addressing_method;
    ppsp_tp_segment_info_t segments<1..*>;
}ppsp_tp_content_info_t;

```

The semantics of ppsp\_tp\_request\_find is extend as follows:

```

Object {
    ppsp_tp_string_t      swarm_id;
    [ppsp_tp_peer_num_t    peer_num;]
    [ppsp_tp_segment_info_t content_info;]
} ppsp_tp_request_find;

```



The semantics of `stream_stats` is extended as follows:

```
Object {
    ppsp_tp_swarm_id_t    swarm_id;
    ppsp_tp_integer_t    uploaded_bytes;
    ppsp_tp_integer_t    downloaded_bytes;
    ppsp_tp_integer_t    available_bandwidth;
    ppsp_tp_content_info_t content_info;
}stream_stats;
```

Currently, the value of `chunk_addressing_method` is identical to the addressing method listed in [section 7.8](#) of [I-D.ietf-ppsp-peer-protocol], as follow:

Method	Description
0	32-bit bins
1	64-bit byte ranges
2	32-bit chunk ranges
3	64-bit bins
4	64-bit chunk ranges
5-255	Unassigned

Table 1: Chunk Addressing Methods

Implementations MUST support "32-bit chunk ranges" (default) and "64-bit chunk ranges". When the `chunk_addressing_method` is 32-bit bins or 64-bit bins, `end_index` in `SegmentInfo` MUST be set to 0. Chunk addressing methods could be extended to allow new algorithms in future specifications, e.g., [BFbitmap]. This document does not extend the semantics and format of Responses.

### 5.3. Compatibility with the Base Tracker Protocol

Trackers are RECOMMENDED to implement extended tracker protocol to be compatible with peers using base tracker protocol or peers using extended tracker protocol. But it is not mandatory. When peers have different implementations with trackers, following 2 catalogs are given:

#### 1. Peer (with extended protocol) vs Tracker (with base protocol)

When peers using extended tracker protocol exchange content information with a tracker only supporting the base tracker protocol, the tracker would respond with 401 (Unsupported version number) `error_code` with an empty message-body (no `peer_addr` and `swarm_result`



attributes), which indicates the messages could not be recognized by the tracker. In this case, peers **MUST** stop interacting with the tracker in extended request messages and use the base tracker protocol instead.

## 2. Peer (with base protocol) vs Tracker (with extended protocol)

The tracker is able to handle all the requests from the peer because all the messages are base tracker protocol messages which could be perfectly accepted by a tracker implementing extended tracker protocol.

### **5.4. Negotiation of Chunk Addressing Methods**

Multiple chunk addressing methods could be used in this document to present content information. But only one of them **MUST** be used for one swarm when a peer communicating with a tracker. Before peers connect to a tracker, they **MUST** get to know the chunk addressing methods supported by the swarm. It is out of scope of the tracker protocol the mechanism used to obtain that information. For example, it could be some out-of-band methods that obtains that information from the web portal, together with other information about the trackers, e.g., IP addresses.

If the chunk addressing method of a swarm can not be supported by a tracker, the tracker is not suggested to serve that swarm. If the chunk addressing method contained in requests is not supported by the swarm controlled by the tracker, the tracker could directly ignore the content related information.





## **5.5. Request/Response Processing**

### **5.5.1. Enhanced FIND Request**

This method allows peers to request to the tracker, whenever needed, a new peer list for the swarm for specific scope of chunks/segments of a media content representation of that swarm.

The peer MUST properly set the request type to FIND, set the PeerID with the identifier of the peer, and set the SwarmID with the identifier of the swarm the peer is interested in. Optionally, in order to find peers having the specific chunks/segments, the peer may include the ContentGroup element in the FIND request message to indicate a specific point in the content timeline.

This message is mainly used for peers in LEECH mode in order to update the peer list of a swarm. For those requests whose peer\_mode are not set to LEECH, the tracker must respond with 400 (Bad request, with reason-phrase "Unknown Messages").

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

The response MAY also include a PeerGroup with PeerInfo data that includes the requesting peer public IP address.

### **5.5.2. 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, "content\_info", 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. The corresponding Response does not need to be extended in this specification.

### **5.5.3. 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 all the swarms (including its status and all content status).

The peer MUST properly form the Request message, set the request type to DISCONNECT, set the PeerID with the identifier of the peer, and randomly generate and set the TransactionID.



## **6. Error and Recovery Conditions**

This document does not introduces 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].

## **7. 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](#)].

## **8. IANA Considerations**

This draft introduces a new version number, see Table 1. Thus corresponding IANA registration is required.

## **9. Acknowledgments**

The authors would like to thank many people for their help and comments, particularly: Zhang Yunfei, Martin Stiernerling, Fei Song, 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.

## **10 References**

### **10.1 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.
- [I-D.ietf-ppsp-base-tracker-protocol] Cruz, R., Nunes, M., Gu, Y., Xia, J., J. Taveira, and Lingli, D., "PPSP Tracker Protocol-Base Protocol (PPSP-TP/1.0)", [draft-ietf-ppsp-](#)



base-tracker-protocol-08 (work in progress), July 2014.

[I-D.ietf-ppsp-peer-protocol] Bakker, A., Petrocco, R. and V. Grishchenko, "Peer-to-Peer Streaming Peer Protocol", [draft-ietf-ppsp-peer-protocol-12](#) (work in progress), June 2014.

## **10.2 Informative References**

[RFC5322] Resnick, P., Ed., "Internet Message Format", [RFC 5322](#), October 2008.

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

[BFbitmap] Bloom, B., "Space/time trade-offs in hash coding with allowable errors.", Communications of ACM Vol. 13, No.7, pp. 422-426, 1970.

### 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

Mario Serafim Nunes  
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

Lingli Deng  
China Mobile  
Email: denglingli@chinamobile.com

