PPSP
INTERNET-DRAFT
Intended Status: Standards Track
Expires: January 5, 2015

Rachel Huang
Huawei
Rui Cruz
Mario S. Nunes
INESC-ID/INOV
Joao P. Taveira
IST/INOV
Lingli Deng
China Mobile
July 4, 2014

PPSP Tracker Protocol-Extended Protocol
draft-huang-ppsp-extended-tracker-protocol-06

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.

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

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 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, which will lead to connection failures. Some P2P applications may require to overcome this shortage of the base tracker protocol. This case requires a message to provide the ability to notify the tracker that a peer has left this tracker.
2. A peer may have the requirement to start streaming the content from some specific point of the content timeline. For example, when the end-user watched only part of a content and decided to

stop and leave, or paused for a long time. When the end-user decides 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.

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

4. Extended Tracker Protocol Overview

The extended Tracker Protocol consists of three Request-Response Extensions (to the CONNECT, 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 disconnect 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). Later the peer DISCONNECTs from the system (step_5).

When in a next time, the end user wants to continue watching the content he/she previously left unfinished, the peer CONNECTs to the corresponding swarm in LEECH mode but sending the specific content information scope.

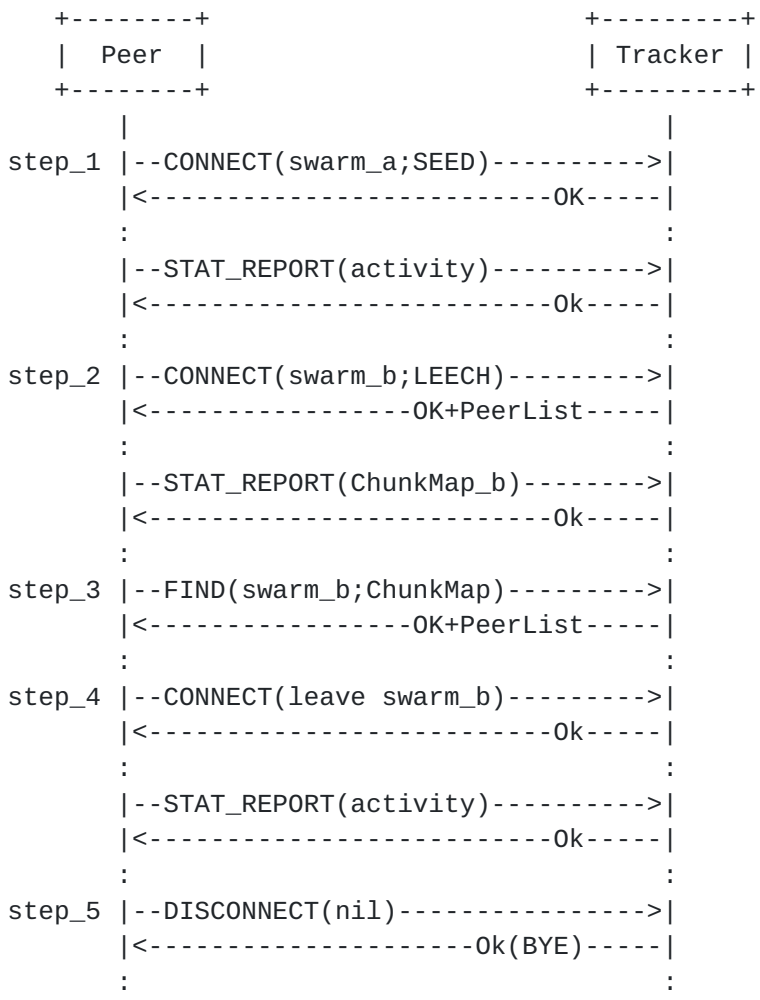

Figure 1: Example of a session for a extended PPSP-TP.

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 2.4.1 of the base tracker protocol [I-D.ietf-ppsp-base-tracker-protocol] except step 5:

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 2.4.2 of the base tracker protocol [I-D.ietf-ppsp-base-tracker-protocol] except step A:

A) At PEER REGISTERED state, if the Peer ID is considered invalid (in the case of a DISCONNECT requests received from an unregistered Peer ID), the tracker responds with either error codes 401 Unauthorized or 403 Forbidden, 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.

```
typedef enum ppsp_tp_request_type {
             PPSP_TP_CONNECT        = 0x02, // or "CONNECT"
             PPSP_TP_FIND           = 0x04, // or "FIND"
             PPSP_TP_DISCONNECT     = 0x06  // or "DISCONNECT"
             PPSP_TP_STAT_REPORT    = 0x08  // or "STAT_REPORT"
} ppsp_tp_request_type_t;
```

```
typedef struct {
           ppsp_tp_version_t           version;
           ppsp_tp_request_type_t      type;
           ppsp_tp_transaction_id_t    id;
           ppsp_tp_peer_id_t           peer_id;
           union {
              struct {
                 ppsp_tp_peer_num_t      peer_num;
                 ppsp_tp_peer_info_t     peer_info;
                 ppsp_tp_swarm_action_t  swarm_actions[];
              } connect;
              struct {
                 ppsp_tp_peer_num_t      peer_num;
                 ppsp_tp_content_info_t  content_info[];
              } find;
              struct {
                 ppsp_tp_stat_t          stats[];
              } stat_report;
        } request_data;
} ppsp_tp_request;
```

The semantics for the content_info element is described as follow:

```
typedef unique_id_t ppsp_tp_segment_start_t;
typedef unique_id_t ppsp_tp_segment_end_t;
typedef unique_id_t ppsp_tp_chunk_addr_t;


typedef struct {
           ppsp_tp_chunk_addr_t      chunk_addressing_method;
           ppsp_tp_segment_info_t    segments[];
} ppsp_tp_content_info_t;


typedef struct {
           ppsp_tp_segment_start_t   start_index;
           ppsp_tp_segment_end_t     end_index; // 0 means no end
} ppsp_tp_segment_info_t;
```

The semantics of Statistics is extended as follows:

```
typedef struct {
           ppsp_tp_stat_type_t       type;
           union {
                  struct {
                     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;
                  } stream_stats;
                  struct {
                     ppsp_tp_content_info_t  content_info[];
                  } content_map;
           } stat_data;
} ppsp_tp_stat_t;
```

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 extends the semantics and format of Responses.

5.3. Extended Request/Response Element in Request Messages

Table 1 specifies the valid string representations for the requests extended in this specification to complement those define in the base tracker protocol.

Extended Request Types	
DISCONNECT	3

Table 1: Extended Request Type of PPSP-TP requests.

The response elements in the extension are identical to those of the base tracker protocol [I-D.ietf-ppsp-base-tracker-protocol].

5.4. 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 using extended tracker protocol exchange content information with a tracker only supporting the base tracker protocol, the tracker could directly ignore the content related information, e.g., ContentGroup element. Peers implementing the extended tracker protocol sending DISCONNECT message to legacy trackers will get responses with 400 (Bad request, with reason-phrase "Unknown Messages"), 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.

5.5. 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.

5.6. Request/Response Processing

5.6.1. 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.

5.6.2. 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, "ContentGroup", 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 has not been extended in this specification.

5.6.3. 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

There are presently no IANA considerations with this document.

9. 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.

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-04 (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-10 (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