PPSP Tracker Protocol – Extended Protocol

draft-huang-ppsp-extended-tracker-protocol-06

PPSP WG

IETF 90 Toronto

Rachel Huang,

Rui Cruz, Mário Nunes,
João Taveira,

Lingli Deng

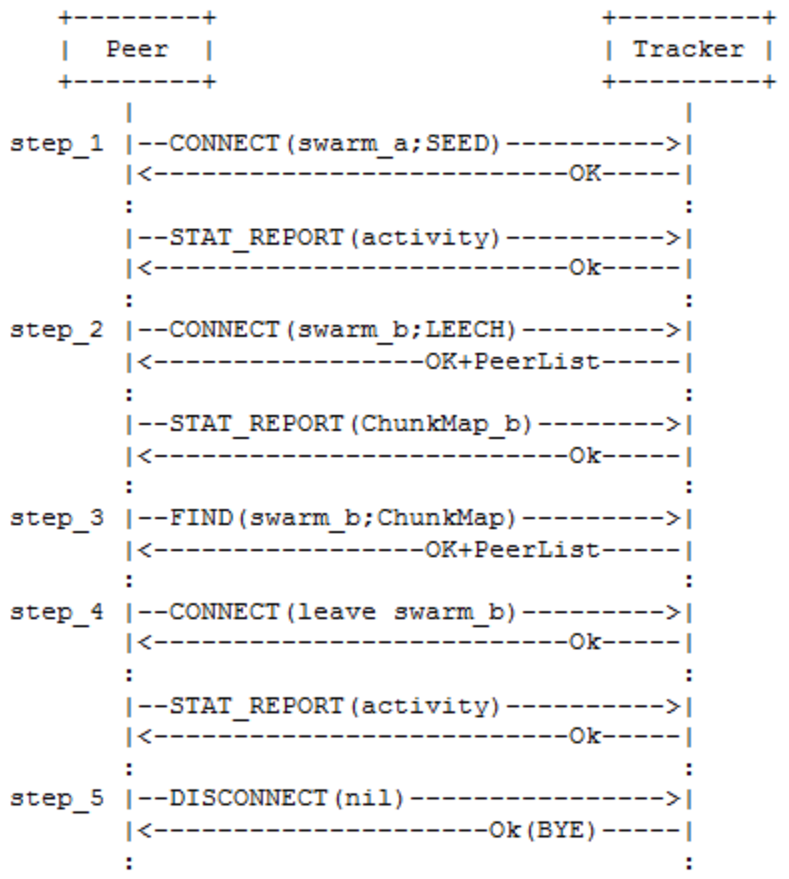
Motivation

- Issues which base tracker protocol may not be able to deal with.
  - Issue 1: Lacking the ability to free resources timely when a peer disconnected from the tracker.
  - Issue 2: Lacking the ability to stream the content from certain specific point.
- Base tracker protocol needs to be extended.

Protocol Design

- Extended “JOIN” and “STAT_REPORT” messages of base tracker protocol.
- Introducing one new optional messages – “DISCONNECT”.
- Consistent with the architecture of the base tracker protocol.
- Retro-compatible with base tracker protocol.
- As a complementary of base tracker protocol.

Extended Tracker Protocol Overview



```
     +--------+                               +---------+
     |  Peer  |                               | Tracker |
     +--------+                               +---------+
         |                                         |
step_1   |--CONNECT(swarm_a;SEED)---------->|
         |<--------------------------OK-----|
         :                                         :
         |--STAT_REPORT(activity)---------->|
         |<--------------------------Ok-----|
         :                                         :
step_2   |--CONNECT(swarm_b;LEECH)--------->|
         |<----------------OK+PeerList-----|
         :                                         :
         |--STAT_REPORT(ChunkMap_b)-------->|
         |<--------------------------Ok-----|
         :                                         :
step_3   |--FIND(swarm_b;ChunkMap)--------->|
         |<----------------OK+PeerList-----|
         :                                         :
step_4   |--CONNECT(leave swarm_b)--------->|
         |<--------------------------Ok-----|
         :                                         :
         |--STAT_REPORT(activity)---------->|
         |<--------------------------Ok-----|
         :                                         :
step_5   |--DISCONNECT(nil)---------------->|
         |<--------------------Ok(BYE)-----|
         :                                         :
```

- 2 Enhanced Messages derived from PPSP-TP/1.0
  - FIND: specific chunks of a content information.
  - STAT_REPORT : content information
- 1 optional messages
  - DISCONNECT: leave the system

Compatibility with Base Tracker Protocol

- Trackers are RECOMMENDED to implement the extended protocol because they can handle peers both using base protocol or extended protocol.
- When a tracker only supporting base protocol
  - Ignore the extended part (content related information) when receiving enhanced messages.
  - Respond with 400 (Bad request) when receiving DISCONNECT message.
  - Peers MUST use base tracker protocol messages instead, when receiving the bad response from the tracker

Chunk Addressing Method (CAM)

- Multiple CAM are supported.
  - identical with peer protocol.
  - Could be extended in the future.
- Only one method MUST be used when peer communicating with tracker.
- Peer MUST obtain the CAM supported by the swarm in advance.
  - How? Out of scope.
- The tracker is NOT RECOMMENDED to serve a swarm when it can’t support the swarm’s CAM.
- If a tracker doesn’t support the CAM in a request, it could directly ignore the content related information.

Message Overview

```
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;
```

```
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;
```

```
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;
```

Next Step

- Ready for adoption?
- Question?

THANK YOU !