PPSP Tracker Protocol

draft-gu-ppsp-tracker-protocol

PPSP WG

IETF 82 Taipei

Rui Cruz (presenter)

Mário Nunes, Yingjie Gu, Jinwei Xia,
David Bryan, João Taveira, Deng Lingli

Main functional entities related with PPSP

- **Client Media Player**
  - is the entity providing a direct interface to the end user at the client device, and includes the functions to select, request, decode and render contents.
  - interfaces with the Peer using request and response mechanisms.
- **Peer**
  - Is a logical entity at the client device embedding the P2P core engine, with a client serving side interface to respond to Client Media Player requests and a network side interface to exchange data and PPSP signaling with Trackers and with other Peers.
- **Tracker**
  - is a logical entity that maintains the lists, as well as the status, of PPSP active peers storing and exchanging chunks for a specific media content.

Terminology

- SEGMENT (of partitioned media)
  - is a resource that can be identified by an ID, an HTTP-URL or a byte-range, and used by a Peer for the purpose of storage, advertisement and exchange among peers.
- SUBSEGMENT (of partitioned media)
  - the smallest unit within segments which may be indexed at the segment level.
- CHUNK
  - is a generic term used to refer to a **SEGMENT** or **SUBSEGMENT** of partitioned streaming media.

Changes since version 5

- This draft corresponds to an enhanced merge of:
  - draft-gu-ppsp-tracker-protocol-05
  - draft-cruz-ppsp-http-tracker-protocol-01
- Includes detailed messages syntax and XML-Schema
- Addresses Authentication & Security aspects based on SASL
- Adds Support for NAT Traversal service via ICE (STUN-Like Tracker)
- Can Support DECADE interoperation
- Is compatible with Distributed trackers organized by RELOAD
- Provides Full PPSP Requirements compliance.
- Changes in messages
  - Removed STAT_QUERY message
  - Re-designed FIND message
  - Re-designed JOIN message

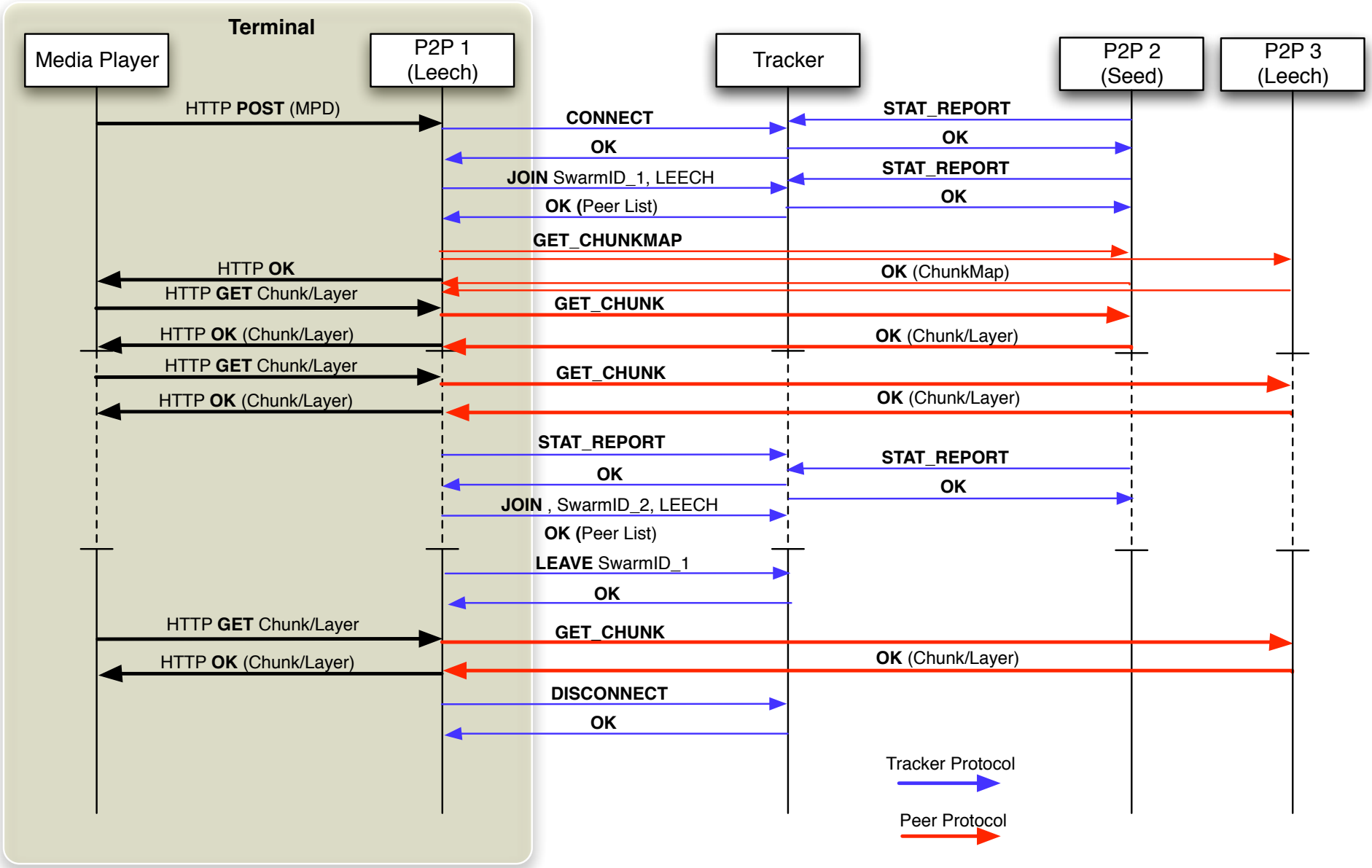
Protocol Design

- The PPSP Tracker Protocol is not used to exchange actual content data with Peers, but information about which Peers can provide which pieces of content.
- The protocol design supports distributed tracker architectures, providing robustness to the streaming service in case of tracker node failure.
- The PPSP Tracker Protocol is a request-response protocol.
  - Requests are sent, and responses returned to these requests.
  - A single request generates a single response.
- The Tracker can provide NAT traversal services (STUN-like Tracker) by discovering the reflexive address of a Peer via PPSP Tracker Protocol messages

Protocol Overview

- To join an existing P2P streaming service and to participate in content sharing, any Peer must locate a Tracker and:
  - Establish a **CONNECT**ion to the system
  - **JOIN** a swarm of Peers streaming a content
  - Obtain or **FIND** a selected List of those Peers
- A Peer can **LEAVE** a swarm but keep active in the P2P streaming service for other swarms
- A Peer sends **STAT-REPORT**s to the Tracker to inform about its status and supply statistic information.
- To terminate all its activity in the P2P streaming service the Peer **DISCONNECT**s for the Tracker.

A Typical PPSP Session



Request Messages

(uploaded, downloaded) left PeerID, the IP addresses

peer IP addresses and link status

PeerID

- The method allows a security layer to be negotiated in the authentication protocol exchange between the Peer and the Tracker

- The method allows a security layer to be negotiated in the authentication protocol exchange between the Peer and the Tracker

Request Messages

VoD

the live streaming modes):

- The tracker adds the Peer to the candidate peers list for
- The joining peer may have none or just some chunks (LEECH), or all the chunks (SEED/LIVESEED) of a content.
  - The type of participation in the swarm is announced and can be SEED, LIVESEED or LEECH
    - The Peer may specify the starting Chunk of a content when joining, restrict the number of candidate peers to receive form the Tracker and provide NAT capabilities.

- The type of participation in the swarm is announced and can be SEED, LIVESEED or LEECH
- The Peer may specify the starting Chunk of a content when joining, restrict the number of candidate peers to receive form the Tracker and provide NAT capabilities.

Request Messages

- Is initiated by the peer, periodically while active.
  - Contains activity statistics.
  - Contains ChunkMaps for all the streaming contents the Peer is currently joined.

contents the Peer is currently joined.

Request Messages

– allows peers to request to the Tracker the peer list for the swarm or for specific chunks

content, restrict the number of candidate peers to receive from the Tracker and provide NAT capabilities.

Request Messages

– used by a Peer to notify the Tracker that it no longer wish to participate in a particular swarm (for both modes): VoD or Live streaming

- The Tracker deletes the corresponding activity records related to the peer.
- The Peer may however continue active in other swarms.

- The Peer may however continue active in other swarms.

Request Messages

- Used when the Peer intends to leave the system and no longer participate in any swarm.
  - The Tracker deletes the corresponding activity records related to the peer (including its status and all content status for all swarms)
  - The Tracker MUST remove the peer from the peer lists and from all swarms the peer was joined.

Messages Syntax

Requests and Responses with XML
Requests and Responses with XML encoded message bodies.

Content-
Content-Type: < <ContentLenght>
< ContentType>
Request_Body>

Content- > <StatusMsg>
Content-Length: <ContentLenght>
Content-Encoding: <ContentType>
< ContentCoding>
Response_Body>

Messages Syntax

PeerID

header with "   ) and authentication token.

- The Response messages MAY use Content-Encoding entity-

gzip

header with "gzip" compression scheme.

<Response>****</Response> <Method Version="#">

<

<AuthToken PeerID>

<TransactionID>***</AuthToken> <!-- on Request except CONNECT-->

</ ...XML information specific of the Method... TransactionID>

ProtocolName

</ProtocolName>

Final Remarks

- Structured Media streaming (SVC/MDC/MVC/multi-bitrate)

- The Media Player application is the entity that should "know" (via a **requester/re-assembler** module) **how** and **what** to **request** (to a Peer) and **decode** the received Structured Media (from the Peer) in order to "prepare" it to **present** to the User.

- The Media Player application is the entity that should "know" (via a **requester/re-assembler** module) **how** and **what** to **request** (to a Peer) and **decode** the received Structured Media (from the Peer) in order to "prepare" it to **present** to the User.

- The Authors would like to ask for the Tracker Protocol defined in this draft to be adopted as PPSP Working Group draft

Comments are welcomed!

THANK YOU !