

# PPSP Tracker Protocol

draft-gu-ppsp-tracker-protocol

PPSP WG

IETF 83 Paris

Rui Cruz (presenter)

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

# Protocol Design

- The PPSP Tracker Protocol is used to control information about which Peers can provide certain content.
- The protocol design supports distributed tracker architectures, providing robustness to the streaming service in case of individual 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 Protocol design does not require Peers to be “connected” to the tracker all the time
- 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 optionally **FIND** selected List of those Peers
- A Peer can **DISCONNECT** from 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.
- **[NEW]** A peer can re-**CONNECT** to the system and implicitly re-**JOIN** all swarms it was previously sharing.

# Request Messages

- **CONNECT:**

- used when a Peer “registers” to the system.
  - The Peer provides its Peer-ID, and the IP addresses (IPv4, IPv6).
  - The Tracker records the Peer-ID, connect-time, peer IP addresses and link status.
  - The method allows a security layer between the Peer and the Tracker.
- [NEW] used, with implicit JOIN semantics when a Peer re-“registers” to the system to resume previous activity in swarms

# Request Messages

- **JOIN:**

- used by a Peer to notify the Tracker that it wishes to participate in a particular swarm (for both VoD or Live streaming modes):
  - The joining peer may have none or just some chunks (LEECH), or all the chunks (SEED) of a content.
  - The tracker checks the PeerMode type (LEECH or SEED) and adds the Peer to the peers list for the swarm.
  - The Peer may specify the starting Chunk of a content when joining, restrict the number of candidate peers to receive from the Tracker and provide NAT capabilities.

# Request Messages

- **STAT\_REPORT:**
  - used by a Peer to inform the Tracker on statistic and status data:
    - Is initiated by the peer, periodically while active.
    - May contains activity statistics.
  - When not including statistics data behaves as a **keep-alive** signal to the tracker.

# Request Messages

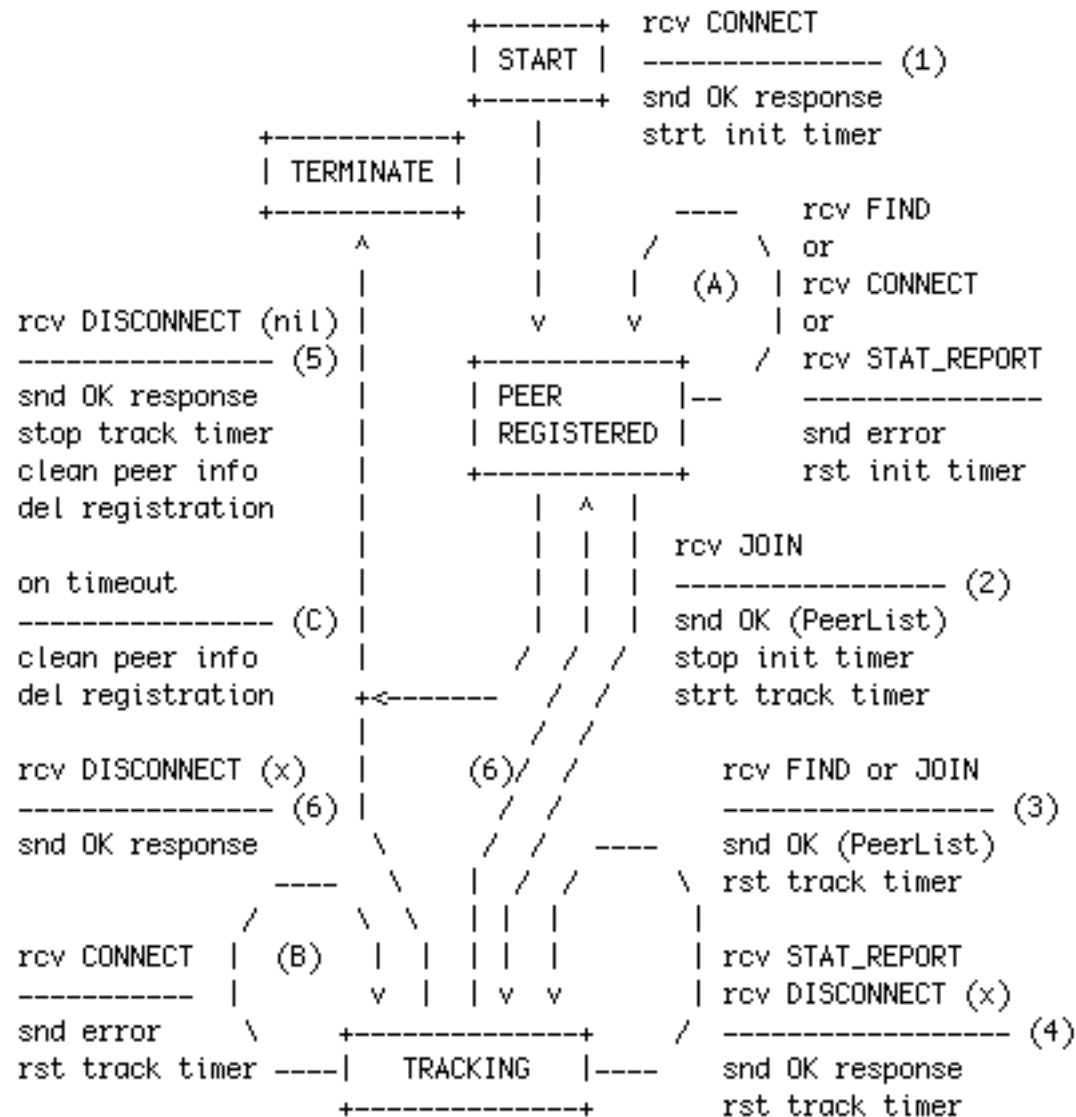
- **FIND:**
  - allows peers to request to the Tracker the peer list for the swarm or for specific chunks of a media content:
    - The Peer may specify the Chunks of interest in a content, restrict the number of candidate peers to receive from the Tracker and provide NAT capabilities.

# Request Messages

- **DISCONNECT:**
  - Used when the Peer intends to leave a **swarm** or **the system** and no longer participate:
    - 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-ID from the peer lists and from the swarms the peer was joined.

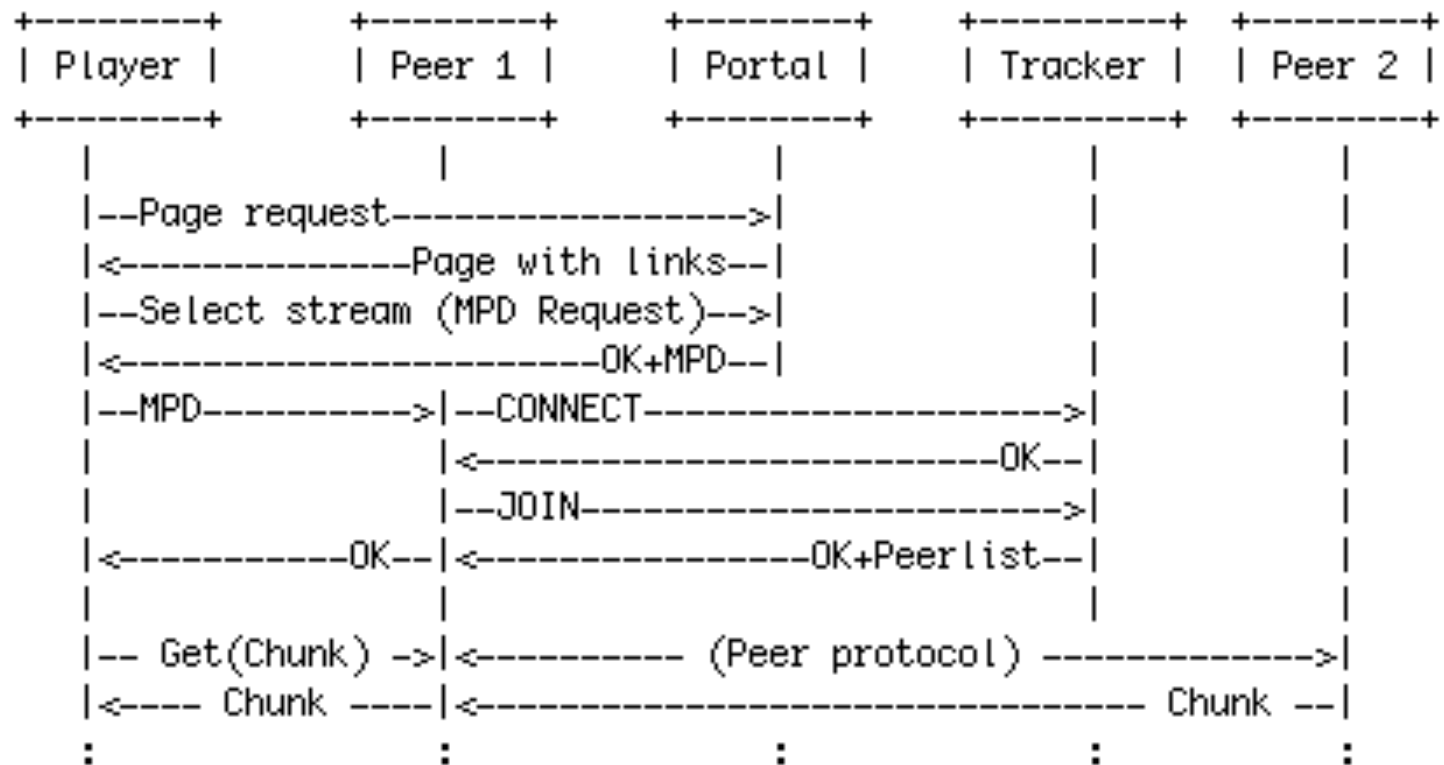


# Tracker STATE MACHINE



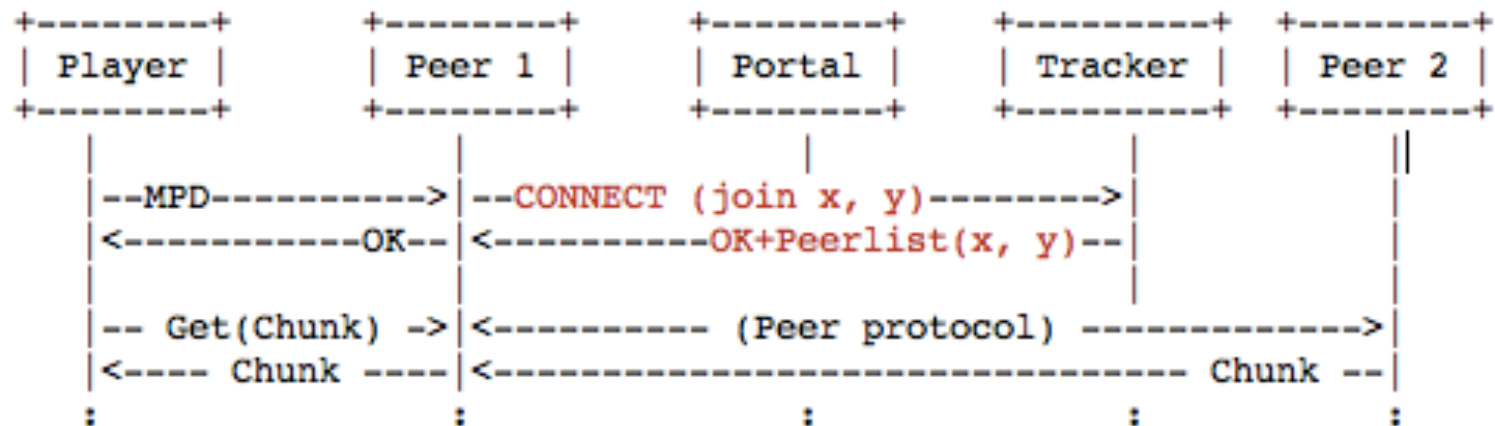
# A PPSP Session

With a peer **CONNECT** to the system and **JOIN**ing a swarm.



# A PPSP Session

[NEW] With a peer re-**CONNECT**ing to the system and implicitly re-**JOIN**ing all swarms it was previously sharing.



# Implementation

- Implemented in **C**
- Available as library “libppsp”

# Changes since version 6

- Includes detailed syntax and semantics of all messages
- Describes the Tracker State-Machine
- Addresses Authentication & Security aspects
- Includes (optional) support for NAT Traversal service via ICE (STUN-Like Tracker)
- Is compatible with Distributed tracker architectures
- Provides Full PPSP Requirements compliance.

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

Comments are welcomed!

**THANK YOU !**