PPSP Tracker Protocol

draft-ietf-ppsp-base-tracker-protocol

PPSP WG

IETF 90 Toronto

Rui Cruz (presenter, remotely)

Rui Cruz, Mário Nunes, Yingjie Gu, Jinwei Xia, Rachel Huang

Deng Lingli, João Taveira

8/21/14

Protocol Overview

- The Base Tracker Protocol uses three messages for the operation:

  - **CONNECT** message to "register" the Peer (on first usage) and to request actions on swarm(s) of streaming contents;

  - **STAT-REPORT** message to periodically inform the Tracker about its status and supply statistic information;

  - **FIND** message to request peer list updates from the Tracker;

- To terminate all its activity in the P2P streaming service the Peer may:

  - Send a **CONNECT** requesting action to LEAVE swarm(s);
  - Stop sending periodic **STAT_REPORT**;

Request Messages

- **CONNECT**:
  - used (on first usage) for Peer to "register" to the system and request "actions" on swarms;
  - if Peer already registered, to request further "actions" on swarms;
    - The Peer provides its Peer-ID, and the IP addresses on its interfaces (IPv4, IPv6).
    - The Peer provides "actions" on one or more swarm(s)
    - The Tracker records/updates the Peer-ID, connect-time, peer IP addresses and link status.
    - The Tracker checks if Peer is LEECH or SEED in swarm "actions" requested, and updates the Peer-ID information to the peer lists.
  - The method allows a security layer to be established between the Peer and the Tracker.

Request Messages

- **STAT_REPORT**:
  - used by a Peer to inform the Tracker on statistic data and status:
    - initiated periodically by the peer, while active in swarms.
    - May contain activity statistics.
  - When not including statistics data, behaves as a **keep-alive** signal to the tracker.

Request Messages

- **FIND**:
  - used by a Peer to request to the tracker an update of peers active in a swarm:
    - initiated by the peer, whenever needed.
    - the Tracker takes peer status, capabilities and priority into consideration (determined by network topology preference, operator policy preference, etc.).
  - includes a **PeerNum** element to indicate the maximum number of peers returned in the list (scoped by the attributes in PeerNum).
  - If no **PeerNum** attributes in the request, a **random sample** from the peer population **is returned**

Changes in -05

- Finalized the formal description of protocol messages using a **C language syntax**:
  - Does not limit the encoding of messages to text-based (XML, JSON, etc.) or Binary formats.
  - Allow implementations to define the best representations of data types.
- Draft completely reviewed to reflect the new formal description
- All implementation examples removed from draft

Examples of C language description but using Generalized Types:

– *ppsp_tp_string_t*

– *ppsp_tp_integer_t*

- In case of implementation in XML
  - a) *ppsp_tp_string_t* -> **String**
    Example of element: "<ASN>AS1234</ASN>"
  - b) *ppsp_tp_integer_t* -> **Integer**
    Example of element: "<Priority>10</Priority>"
    Example of attribute: "... priority='10'"

- In case of implementation as a char array
  - *ppsp_tp_string_t* -> **char ***
    Example: "AS1234\0"

- In case of implementation as uintXX_t
  - *ppsp_tp_integer_t* -> **uint8_t**
    Example for Priority value: 0x0A
  - *ppsp_tp_integer_t* -> **uint16_t**
    Example for Address Port value: 8088

Next Steps

- **Move Draft to WGLC?**