

PPSP
Internet Draft
Intended status: Informational
Expires: September 5, 2016

Hongke Zhang
Fei Song
Di Wu
Mi Zhang
Tianming Zhao
Beijing Jiaotong University
March 4, 2016

Usage of the Peer-to-Peer Streaming Protocol (PPSP)
draft-zhang-ppsp-usage-04.txt

Abstract

This document focuses on several crucial operation issues of Peer-to-Peer Streaming Protocol (PPSP) usage, considering two basic modes: Leech mode and Seed mode. The settings of related parameters for default PPSP scenario reference to tracker protocol and peer protocol respectively. In addition, the limitations and gaps of current PPSP system are identified at with the standpoint of satisfying PPSP requirements.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 5, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Terminology	3
3.	Operation of PPSP System.....	3
3.1.	Operation Overview.....	4
3.2.	Operation Illustration.....	4
4.	Suggestions for Parameters Setting in PPSP System	9
4.1.	Parameters Setting in Tracker Protocol	10
4.2.	Parameters Setting in Peer Protocol	10
5.	Limitations and Gaps Analysis	11
6.	Security Considerations.....	12
7.	IANA Considerations	12
8.	References	12
8.1.	Normative References.....	12
8.2.	Informative References.....	13
9.	Acknowledgments	13

1. Introduction

The Peer-to-Peer Streaming Protocol (PPSP) supports both live and Video on Demand (VoD) streaming. It consists of two basic protocols: the PPSP peer protocol [[RFC7574](#)] and the PPSP tracker protocol [[I-D.ietf-ppsp-base-tracker-protocol](#)]. Both of them are proposed from individual perspective based on PPSP structure. However, it is unnecessary for end users to understand the whole procedure works and how to set the parameters when combining above two mentioned protocol together in application. More importantly, the potential limitations of current protocol should also be learnt and known to the community.

The tracker protocol, in a request/response model, handles the initial and periodic exchange of meta-information between trackers and peers. The PPSP Tracker Protocol provides communication between trackers and peers, by which peers send meta information to trackers, report streaming status and obtain peer lists from trackers.

This document introduces several crucial operation issues in usage, working in two basic modes: Leech mode and Seed mode. Setting of Related parameters for default PPSP scenario is also given by tracker protocol and peer protocol respectively. In addition, the limitations and gaps of current PPSP system are identified from the standpoint of satisfying PPSP requirements.

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](#)].

The document makes extensive use of the terminology and definitions inherited from Concepts and Terminology for PPSP peer protocol [[RFC7574](#)] and PPSP-TP/1.0 [[I-D.ietf-ppsp-base-tracker-protocol](#)] in this document.

3. Operation of PPSP System

Different with previous protocol-related drafts, the operation process of PPSP system in this document focuses on how to make multiple entities working together, such as peers, trackers, portals, etc., and achieve corresponding functions. Both macroscopic overview and specific steps are provided in the following sections.

3.1. Operation Overview

The PPSP supports both real-time and on-demand streaming modes, which consists of two protocols: the PPSP tracker protocol and the PPSP peer protocol.

The tracker refers to a directory service that maintains a list of active peers participating in a specific audio/video channel or in the distribution of a streaming file. It is a logical entity, which can be deployed centralized or distributed, and in this document, it is treated as a single logical entity.

The peer refers to a participant in a P2P streaming system that not only receives streaming content, but also caches and streams streaming content to other participants. Based on the properties of peers, there are two different modes (Leech mode and Seed mode) in PPSP. These two modes will be detailed in [Section 3.2](#).

The basic communication unit of PPSP is message. In peer protocol, multiple messages are typically multiplexed into a single datagram in transmission process. And in the PPSP system, there are several rules MUST be obeyed.

1. In the same swarm, peers MUST use the same chunk addressing method to ensure that peers can communicate with each other smoothly.
2. The portal needs to select an appropriate tracker supporting the same encoding type as the peer. Besides, the portal needs to distinguish the VoD content from live content and then selects the appropriate tracker for peers.

3.2. Operation Illustration

The normal operation process of the PPSP system is illustrated in Figure 1. The related entities and elements are described as follows:

Tracker: A logical entity that provides the peer list to peers.

Portal: A logical entity that provides the Media Presentation Description (MPD) files to peers.

Peer A: A peer that has content resource and wants to share it with others. (PeerMode is of Seed)

Peer B: A peer that wants to join swarm x to obtain the content provided by Peer A. (PeerMode is of Leech)

Tracker	Portal	Peer A	Peer B	Peer C	Peer D
<-CONNECT(Join Swarm x)					
-----OK----->					
<----STAT_REPORT-----					
-----OK----->					
:	:				
<-----Select Swarm x----					
-----OK+MPD(x)----->					
<-----CONNECT(Join Swarm x)-----					
-----OK+PeerList----->					
:	:				
<-HANDSHAKE-					
--HS+HAVE-->					
<-PEX_REQ---					
--PEX_RES-->					
<-HANDSHAKE->					
<-----STAT_REPORT-----					
-----OK----->					
:	:				
<-----HS+HAVE+CHOKE-----					
<--REQUEST--					
---DATA---->					
<--ACK, HAVE-					
:	:				
<-----STAT_REPORT-----					
-----OK----->					
<-----UNCHOKE-----					
<-----REQUEST----->					
:	:				
<-----DATA-----					
<-----ACK, HAVE----->					
:	:				
<---HAVE-----					
<---HAVE---->					
<---REQUEST--					
<-----REQUEST-----					
<---DATA---->					
<-----DATA----->					
:	:				
<-KEEPALIVE-					
<-KEEPALIVE->					
<-----KEEPALIVE----->					
<-----STAT_REPORT-----					
-----OK----->					
<-HANDSHAKE-					
<-HANDSHAKE->					
<-----HANDSHAK----->					
<---CONNECT/FIND(Leave Swarm x)-----					
<---CONNECT/FIND(Join Swarm y,z)-----					

Figure 1 Procedures of PPSP System

Peer C (Peer D): A peer that wants to join swarm x to obtain the content provided by Peer A. And it has finished part of the content transmission. (PeerMode is of Leech)

Assume that Peer A (Seeder) who wants to share a static/dynamic video content with other peers. Firstly, Peer A MUST send a CONNECT message to a tracker to start/join swarm x.

After a correct CONNECT message is received, the tracker responses to Peer A with an OK message.

In order to stay in swarm x, Peer A should send the STAT_REPORT message to the tracker periodically. Normally, it is recommended to set 3 minutes as the value of Track_timeout (More details described in [section 4](#)). An OK message should be generated and sent back to Peer A whenever STAT_REPORT message reaches to the tracker.

Assume that Peer B (Leecher) wants to watch this video content provided by Peer A. Peer B needs to connect and login a service Portal with its peer ID to get the MPD file, includes the IP address(es) of tracker(s) and swarm x's ID., of the selected swarm x

Then Peer B starts to communicate with the tracker and try to join the swarm x by sending a CONNECT message to the tracker. Such behavior will trigger the tracker to send response back to Peer B with an OK+PeerList message if previous check was correct. The message gives Peer B a proper list including peers' name and IP addresses (only Peer A and its address here).

Until now, Peer B knows which peer (Peer A here) has been in the swarm x already. It sends a datagram with HANDSHAKE message to Peer A (as only one seeder in the swarm x). The payload of the HANDSHAKE message is a channel ID and a sequence of protocol options.

Then Peer A decides whether to communicate with Peer B or not, base on its status and current network capacities. Once Peer A decides to respond, it returns a datagram with HANDSHAKE+HAVE message to Peer B. (HS is the abbreviation of HANDSHAKE in Figure 1)

After acquiring the acknowledgement of Peer A, Peer B may use another way (sending PEX_REQ message to Peer A) to update PeerList as OPTIONAL. This message could help Peer B to discover other new peers, which could not be provided by the tracker.

Peer A returns a datagram with PEX_RES message. Assume that the information of Peer C and Peer D is included in it.

As the rules mentioned before, if Peer B wants to initial a new conversation with Peer C or D, it MUST send a datagram containing HANDSHAKE message.

Similar with the situation of Peer A, Peer C or D needs to decide whether to respond to Peer B or not. Assume that Peer C is willing to communicate with Peer B. Thus, it sends back a datagram containing HANDSHAKE+HAVE message. If Peer D wants to deny Peer B, it MUST send a datagram including the HANDSHAKE+HAVE+CHOKe message.

On receiving previous datagram, Peer B checks the messages and knows which is available. Then it can send datagrams containing the REQUEST message to Peer A and C asking for chunks.

After Peer A or C receives the Peer B's request, it SHOULD send data to Peer B. The datagram content depends on the video type: INTEGRITY+DATA message for static video and SIGNED_INTEGRITY+DATA message for dynamic video.

Upon receiving the corresponding data, Peer B sends back a datagram including an ACK message to Peer A and C. Peer B SHOULD also send a datagram containing HAVE message to all other peers that in the swarm x for announcement purpose. The timing to send HAVE message depends on Peer B.

In order to avoid timeout of track timer, Peer B MUST send STAT_REPORT message to the tracker. Such report is confirmed when the tracker's OK message reaches Peer B.

For demonstrating all the functionalities, Peer D is supposed to release previous rejection for Peer B by sending an UNCHOKe message.

Then, Peer B can send a new REQUEST message to Peer D.

Peer D responses with the actually data message. After its content integrity is verified, Peer B MAY send HAVE message to other peers in swarm x.

Peer C and D can also ask Peer B for chunks by sending REQUEST message. Corresponding chunks could be sent depends on Peer B

If the above peers want to stay in the swarm, they need to send the STAT_REPORT message to the tracker and send the KEEP_ALIVE message to other peers periodically.

After successfully received all the necessary content, Peer B can close the connection by sending a HANDSHAKE message to all peers in

swarm x. An all 0-zeros channel ID MUST be embedded in HANDSHAKE message. Meanwhile, Peer B SHOULD send STAT_REPORT or CONNECT message to log out and eliminate its state machine in tracker.

Peer B MAY use CONNECT message to join a new swarm, such as swarm y or z in Figure 1. Similar instruction mentioned before can be utilized for data exchanging.

Useful Message List:

CONNECT message

This message is used to register/leave a PPSP system and request swarm actions with tracker.

FIND message

This message is used to request a new peer list to tracker whenever needed. It is also used when a peer wants to leave the PPSP system with tracker.

STAT_REPORT message

This message is used to send status and statistic data to tracker, in order to facilitate the tracker service. This message MUST be periodically sent while the peer is active.

OK message

This message is used for tracker to convey that has successfully received the last message.

OK+PeerList message

This message is used for tracker to respond proper PeerList to peer.

HANDSHAKE message

This message MUST be sent as the first message in the first datagram between peers, in order to start communication between peers.

HAVE message

This message is used to convey which chunks a peer has available for download.

DATA message

This message is used to transfer chunks of content.

ACK message

This message is used to acknowledge received chunks after receiving a DATA message.

REQUEST message

This message is used to request one or more chunks from another peer.

INTEGRITY message

This message carries information required by the receiver to verify the integrity of a chunk. It is usually used in static content.

SIGNED_INTEGRITY message

This message is used to verify chunks in live streaming.

CHOKe message

The message is used to inform another peer that it will no longer respond to any REQUEST messages from that peer.

UNCHOKe message

This message is used to inform another peer that it will respond to new REQUEST messages from that peer again.

PEX_REQ & PEX_RES messages

These message allows peers to exchange the transport addresses of the peers they are currently interacting with, thereby reducing the need to contact a central tracker.

KEEPALIVE message

This message SHOULD be sent periodically to each peer it wants to interact with in the future.

4. Suggestions for Parameters Setting in PPSP System

In the procedure of constructing the PPSP system, parameters setting is quite important. This section will discuss related issues in tracker protocol and peer protocol. The default values are provided

as references. The practical setting can be adjusted according to different scenarios.

[4.1. Parameters Setting in Tracker Protocol](#)

Table 1 shows parameters, their default values and description in the PPSP tracker protocol.

Name	Default	Description
Init_timeout	30 seconds	Maximum value of the "init timer" used in the "per peer transaction state machine"
Track_timeout	3 minutes	Maximum value of the "track timer" used in the "per peer transaction state machine"
STAT_REPORT Period	3 minutes	Maximum period of STAT_REPORT message
Retry_timeout	3 minutes	Maximum waiting time until a peer initiates a retry process
ConcurrentLinks	NORMAL	Concurrent connectivity level of peers, HIGH, LOW or NORMAL
OnlineTime	NORMAL	Availability or online duration of peers, HIGH or NORMAL
UploadBWlevel	NORMAL	Upload bandwidth capability of peers, HIGH OR NORMAL

Table 1 PPSP Tracker Protocol Defaults

[4.2. Parameters Setting in Peer Protocol](#)

Since the PPSP peer protocol has a detailed description about parameters, this section only adopt it as a reference to summarize Table 2, which reveals some of the parameters with their default values and descriptions. Some parameters should be recommended as a fixed value, and others could alter according to users' demands and network conditions.

Name	Default	Description
Chunk Size	var 1024 bytes recommended	(Maximum) Size of a chunk
Static Content Integrity Protection Method	1 (Merkle Hash Tree)	Methods for protecting the integrity of static content
Live Content Integrity Protection Method	3 (Unified Merkle Tree)	Methods for protecting the integrity of static content including "sign all" and "Unified Merkle Tree"
Merkle Hash Tree Function	0 (SHA1)	Hash function used for the Merkle Hash Tree
Live Signature Algorithm	13 (ECDSAP2 56SHA256)	Must be defined for live streaming
Chunk Addressing Method	2 (32-bit chunk ranges)	Methods of chunk addressing
Live Discard Window	var	Must be defined for live streaming
NCHUNKS_PER_SIG	var	Must be defined in the Signed Munro Hash
Dead peer detection	No reply in 3 minutes + 3 datagrams	Guideline for declaring a peer is dead
KEEPALIVE Period	2 minutes	Maximum period for a peer to send KEEPALIVE datagram to other peers

Table 2 PPSP Peer Protocol Defaults

5. Limitations and Gaps Analysis

This section aims to identify the limitations and gaps of the current PPSP system from the standpoint of satisfying PPSP requirements.

- One target of PPSP is extending current Peer-to-Peer (P2P) system in mobile and wireless environments [[RFC6972](#)]. However, the message used in PPSP system does not contain related information such as the packet loss rate and battery status, which is essential for wireless and mobile environments.

2. The PPSP system provides two ways to fetch the PeerList. Peers can obtain the PeerList from the tracker or they can get it through the PEX_REQ and PEX_RES messages. When both methods are available, how to update the local PeerList efficiently is still not clear.
3. The STAT_REPORT message of tracker protocol does not support the exchanges of content data information, like chunk maps, between an active peer and a tracker. Thus, whenever a new peer wants to join a swarm, the relevant tracker could only use PeerMode to choose the PeerList and return to the new peer. In some cases, there may be only one seeding peer, while several peers that already finished part of the content transmission and are willing to share with others. As a result, the tracker could not provide the high quality PeerList but just one seeder. Thus, the peer could only rely on using the PEX-REQ message to update PeerList.
4. In some cases, the user may want to adjust the video definition based on the bandwidth (or user demand) automatically (or manually). Or the user may watch videos and play online games at the same time, and he/she don't want the videos occupy too much of the bandwidths. This is adaptive multi-rate control for both users and ISPs. Rather than letting the ISPs or governments limit the download links, why not we add some controllable limits in the protocol.
5. For safety and management reasons, PT (private tracker) has become popular in recent years. It is not sure if this should be taken into consideration in PPSP. If the answer is positive, then tracker protocol should make some alters in finding & connecting private tracker and add data traffic statistics part.

6. Security Considerations

This document does not contain any security considerations.

7. IANA Considerations

There are presently no IANA considerations with this document.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

8.2. Informative References

- [RFC7574] Bakker, A., Petrocco, R., and V. Grishchenko, "Peer-to-Peer Streaming Peer protocol (PPSP)", [RFC 7574](#), October 2015.
- [I-D.ietf-ppsp-base-tracker-protocol] Cruz, R., Nunes, M., Gu, Y., Xia, J., and J. Taveira, "PPSP Tracker Protocol-Base Protocol (PPSP-TP/1.0)", [draft-ietf-ppsp-base-tracker-protocol-12](#) (work in progress), January 2016.
- [RFC6972] Zhang, Y. and N. Zong, "Problem Statement and Requirements of the Peer-to-Peer Streaming Protocol (PPSP)", [RFC 6972](#), July 2013.

9. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Hongke Zhang
Beijing Jiaotong University (BJTU)
Beijing, 100044, P.R.China

Email: hkzhang@bjtu.edu.cn

Fei Song
Beijing Jiaotong University (BJTU)
Beijing, 100044, P.R.China

Email: fsong@bjtu.edu.cn

Di Wu
Beijing Jiaotong University (BJTU)
Beijing, 100044, P.R.China

Email: diwu2@seas.upenn.edu

Mi Zhang
Beijing Jiaotong University (BJTU)
Beijing, 100044, P.R.China

Email: 13120174@bjtu.edu.cn

Tianming Zhao
Beijing Jiaotong University (BJTU)
Beijing, 100044, P.R.China

Email: 14125070@bjtu.edu.cn