

PCP WG
Internet-Draft
Intended status: Informational
Expires: November 5, 2012

M. Boucadair, Ed.
France Telecom
T. Zheng
P. NG Tung
X. Deng
J. Queiroz
Orange Labs
May 4, 2012

Behavior of BitTorrent service in PCP-enabled networks with Address Sharing

draft-boucadair-pcp-bittorrent-00.txt

Abstract

This document describes the behavior of BitTorrent service in the context of PCP-enabled address sharing functions. It provides an overview of the used testbed and main results of the tests that have been conducted in order to assess the limitations of an architecture based on shared IP addresses.

Status of this Memo

This Internet-Draft is submitted to IETF 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 November 5, 2012.

Copyright Notice

Copyright (c) 2012 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

Internet-Draft

BitTorrent & PCP

May 2012

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.	BitTorrent Overview	3
2.1.	BitTorrent at a Glance	3
2.2.	Software Configuration	4
2.2.1.	BitTorrent Client	4
2.2.2.	BitTorrent Server	4
2.2.3.	BitTorrent Tracker	4
3.	Testbed Overview	4
3.1.	Testbed Description	4
3.2.	Files	5
3.3.	Methodology	5
4.	Description of Tests	6
4.1.	Connection to Overlay Test Group	6
4.2.	Upload Test Group	7
4.3.	Mutual Download Test Group	7
4.4.	Simultaneous Download Test Group	8
5.	Results	11
5.1.	Allow Same IP Address & PCP Disabled	11
5.2.	Forbid Same IP Address & PCP Disabled	13
5.3.	Allow Same IP Address & PCP Enabled	15
5.4.	Forbid Same IP Address & PCP Enabled	17
6.	Conclusions	19
7.	IANA Considerations	20
8.	Security Considerations	20
9.	References	20
9.1.	Normative References	20
9.2.	Informative References	21
	Authors' Addresses	21

1. Introduction

Recently, several proposals have been disseminated within IETF to allow for IPv4 service continuity. These solutions share the same IP address among several subscribers (e.g., CGN (Carrier Grade NAT) [[I-D.ietf-behave-lsn-requirements](#)] or A+P [[RFC6346](#)])

Several issues are encountered in address sharing context as elaborated in [[RFC6269](#)].

This memo focuses on BitTorrent as an example of application which applies a restriction based on IP address. This memo describes a testing campaign that has been carried out to assess the impact of IP shared address on BitTorrent.

A particular focus has been put on the impact of activating port forwarding (using PCP [[I-D.ietf-pcp-base](#)]) on the download speed.

2. BitTorrent Overview

2.1. BitTorrent at a Glance

BitTorrent is a distributed file sharing infrastructure. It is based on P2P (Peer to Peer) techniques for exchanging files between connected users. Three parties are involved in a BitTorrent architecture as detailed hereafter:

1. The Server: The server into which, has been uploaded the torrent file.
2. The Tracker: Maintains a list of clients which have the file or some portions of that file.
3. The Client: Entities which are downloading and/or uploading portions of the file. Two categories of clients may be

distinguished:

- A. Leechers: Clients which are currently downloading the file but do not yet detain all the portions of the file. As for the portions already obtained, the leechers upload them towards requesting clients;
- B. Seeders: Clients which detain all the portions of the file and are uploading them to other requesting clients.

A torrent file is a file which includes the meta-data information of the file to be shared: the file name, its length, a hash and the URL

of the tracker. In order to download a given file, a BitTorrent client needs to obtain the corresponding torrent file. Afterwards, it connects to the tracker to retrieve a list of leechers and seeders. Then, the client connects to those machines and downloads the available portions of the requested file. It uploads also the portions already obtained towards requesting clients.

[2.2.](#) Software Configuration

This section provides an overview of installed tools.

[2.2.1.](#) BitTorrent Client

Various BitTorrent clients are available for public use. The following one has been installed for the purposes of our testing activities:

URL: www.bittorrent.com

[2.2.2.](#) BitTorrent Server

The BitTorrent server that has been used is the following:

URL: www.torrentbox.com

[2.2.3.](#) BitTorrent Tracker

The BitTorrent tracker that has been used is the following:

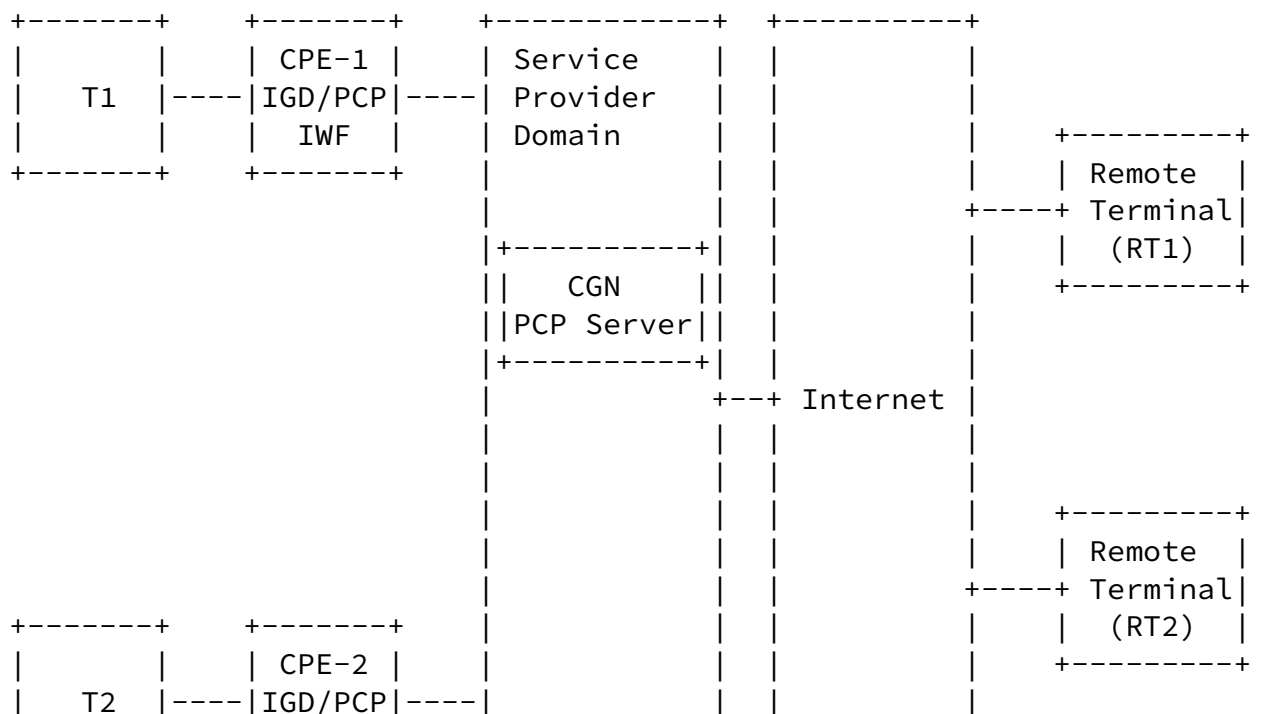
3. Testbed Overview

3.1. Testbed Description

The testbed used to conduct the testing activities is illustrated in the figure below:

- o The CGN DS-Lite which is responsible to share the same IP address among several subscribers. The CGN embeds a PCP Server.
- o CPE-1 and CPE-2 are two CPEs sharing the same IP address (by the CGN). Each CPE embeds a IGD/PCP IWF [[I-D.ietf-pcp-upnp-igd-interworking](#)].
- o T1 (respectively T2) is a machine located in the LAN behind CPE-1 (respectively CPE-2). No NAT is enabled in CPE-1 and CPE-2.

- o RT1 and RT2 are remote machines reachable through Internet. RT1 and RT2 are assigned with public IP addresses.



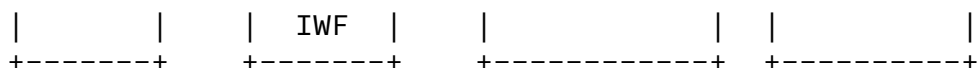


Figure 1: Testbed Overview

3.2. Files

The following table lists the files available in each machine:

Machine' s name	Available files
T1	TestCaenF1 and TestCaenFa
T2	TestCaenF1 and TestCaenFb
RT1	TestCaenFRT1 and TestCaenFRTa
RT2	TestCaenFRT1 and TestCaenFRTb

Table 1: Available files

3.3. Methodology

BitTorrent client can be configured to accept multiple connections using the same IP address. A dedicated parameter can therefore be positioned. This parameter is called: `bt.allow_same_ip`. Possible values that can be taken by this parameter are: FALSE (0) or TRUE

(1).

Tests are conducted using four configurations:

Configuration	bt.allow_same_ip	PCP
Section 5.1	TRUE in all machines (T1, T2, RT1, RT2)	Disabled
Section 5.2	FALSE in all machines (T1, T2, RT1, RT2)	Disabled
Section 5.3	TRUE in all machines (T1, T2, RT1, RT2)	Enabled

Section 5.4	TRUE in all machines (T1, T2, RT1, RT2)	Enabled
-----------------------------	--	---------

When PCP is disabled, all port forwarding entries are flushed out.

4. Description of Tests

This section lists the tests that have been conducted.

4.1. Connection to Overlay Test Group

This table lists the test to assess the ability of distinct machines having the same IP address to connect to BitTorrent overlay.

Test Index	Test Title	Purpose & Description
Test_1	Connection to BitTorrent Overlay	Check if two terminals, having the same public IP address, are able to connect to BitTorrent overlay network. Check if BitTorrent client installed on T1 and T2 machines are able to use the same tracker and that no problems are experienced to use the same tracker by T1 and T2.

Connecting to Overlay Test Group

4.2. Upload Test Group

This test group aims at checking if upload operations are not impacted/restricted due to the presence of several machines with the same IP address.

Test	Test Title	Purpose & Description
------	------------	-----------------------

Index		
Test_2	Uploading distinct files using the same BitTorrent tracker and server	Check if two terminals, having the same public IP address, are able to upload torrent files (referring to distinct files) using the same tracker and same server. Check if torrent files may be uploaded from T1 and T2 using the same tracker. On T1 (resp. T2), generate a torrent file TestCaenFa.torrent (resp. TestCaenFb.torrent) referring to the file TestCaenFa (resp. TestCaenFb) and pointing to the tracker TRA. From T1 (resp. T2) try to put TestCaenFa.torrent (resp. TestCaenFb.torrent) onto server S. Check if the upload operation has succeeded
Test_3	Uploading torrent files referring to the same file	Check if two terminals, having the same public IP address, are able to upload torrent files, which refer to the same file, using the same tracker. On T1 (resp. T2), generate a torrent file TestCaenF1.torrent (resp. TestCaenF1.torrent) referring to the file TestCaenF1 and pointing to the tracker TRA. From T1 (resp. T2) try to put TestCaenF1.torrent (resp. TestCaenF1.torrent) onto server S. Check if the upload operation has succeeded

Upload Test Group

[4.3.](#) Mutual Download Test Group

The purpose of this test group is to check if mutual downloading operations can occur between machines having the same IP address.

Test Index	Test Title	Purpose & Description
Test_4	Mutual Downloading between machines sharing the same IP address	Check if two terminals having the same public IP address can download a file from each another. Check if T1 can download the file uploaded by T2 (ref. Test_2) and vice versa. Three scenarios are to be tested: (1) T1 downloads TestCaenFb but T2 does not download any file from T1, (2) T2 downloads TestCaenFa but T1 does not download any file from T2, (3) T1 downloads TestCaenFb and T2 downloads TestCaenFa at the same time
Test_5	Mutual Downloading between machines located behind an address sharing function	Check if two terminals located behind an address sharing function but assigned with distinct public IP addresses can download a file from each another. Check if T1 can download the file uploaded by T2 (ref. Test_2) and vice versa.

Mutual Download Test Group

[4.4.](#) Simultaneous Download Test Group

This test group aims at checking if simultaneous downloading operations from remote seed(s)/leecher(s) can be performed by several machines sharing the same IP address.

Test Index	Test Title	Purpose & Description
Test_6	Downloading distinct files	Check if two terminals, having the same public IP address, are able to download distinct files available on BitTorrent infrastructure. Check if distinct files available on BitTorrent infrastructure may be downloaded by T1 and T2 simultaneously
Test_7	Downloading the same file located on several seeders	Check if two terminals, having the same public IP address, are able to download the same file located on several seeders. Check if a file available on several seeders may be downloaded from T1 and T2 simultaneously. As an example, check if T1 and T2 can download the same file located in RT1 and RT2 (referred to as TestCaenFRT1)
Test_8	Download the same file available on a single machine	Check if two terminals having the same public IP address are able to download, at the same time, the same file available on a single seed. Check if T1 and T2 can download the same file uploaded by RT1 (referred to as TestCaenFRTa) concurrently. In case the test fails, one of the two host is called the "waiting client"

Internet-Draft

BitTorrent & PCP

May 2012

Test_9	Simultaneous downloading from the same seeder	Check if it is not precluded that a different file can be downloaded by the waiting client from the same seeder. In case Test_7 fails, check that it is not precluded that a different file can be downloaded by the waiting client (T1 or T2) from the same seeder (RT1) at the same time the other terminal (respectively T2 or T1) is downloading TestCaenFRTa. Execute Test_7 in launching on T1 the downloading of TestCaenFRT1 and just few seconds afterwards in launching on T2 the downloading of TestCaenFRT1 and TestCaenFRTa. Check that while T1 is downloading TestCaenFRT1 that does not preclude T2 to concurrently download TestCaenFRTa.
Test_10	Downloading distinct files from the same seeder	Check if the two terminals having the same public IP address are able to download at the same time two distinct files from the same seeder. Check if T1 (respectively T2) can download files uploaded by RT1 (referred to as TestCaenRF1 and TestCaenFRTa) concurrently. Particularly, check if T1 can download TestCaenFRT1 and T2 can download TestCaenFRTa simultaneously
Test_11	Download the same file located on machines having the same IP address	Check if the same file can be downloaded by a given machine from seeders having the same IP address. In RT1, launch the downloading of TestCaenF1. Check that RT1 is downloading portions of TestCaenF1 at the same time from T1 and T2

Test_12	Automatic query to download the same file available on a single machine	Check if the terminal which was waiting can finally download the file once the other terminal has finished. In case Test_7 fails, check that the terminal which was waiting can finally download the file once the other terminal has finished
---------	---	--

Test_13	Download distinct files from two machines having the same IP address	Check if distinct files can be downloaded by the same machine from seeders having the same IP address. Check if RT1 can download simultaneously TestCaenFa (from T1) and TestCaenFb (from T2)
---------	--	---

Simultaneous Download Test Group

5. Results

The following tables summarize the results of the tests listed in [Section 4](#) as performed using the testbed described in [Section 3](#). Four configurations have been tested as documented in [Section 3.3](#).

5.1. Allow Same IP Address & PCP Disabled

The following table summarizes the results of the tests when "bt.allow_same_ip == TRUE" in all involved BitTorrent clients and PCP is disabled.

Index	Results	Downloading Speed
Test_1	No problems have been experienced	
Test_2	Both T1 and T2 are able to upload	

	distinct torrent files using the same tracker and the same server.	
Test_3	Only one machine can upload a torrent file referring to the same file. The server ensures that only one single torrent file corresponding to the same file is listed in its base.	

Test_4	Three scenarios have been tested: (1) T1 downloads TestCaenFb but T2 does not download any file from T1 (2) T2 downloads TestCaenFa but T1 does not download any file from T2 (3) T1 downloads TestCaenFb and T2 downloads TestCaenFa in the same time. For all these scenarios, mutual downloading between T1 and T2 is not observed.	
Test_5	No mutual downloading between T1 and T2 was observed.	
Test_6	Both T1 and T2 are able to download distinct files from the BitTorrent infrastructure.	T1: 50-110KBps; T2: 60-80KBps
Test_7	Both T1 and T2 are able to download the same file located in several seeders. Mutual downloading between T1 and T2 is not observed.	T1 and T2: 50-70KBps
Test_8	Both T1 and T2 are able to download	T1:

	TestCaenFRTa from RT1 simultaneously. Mutual downloading between T1 and T2 is not observed.	50-70KBps; T2: 40-80KBps
Test_9	Not applicable	
Test_10	No problem has been encountered. Distinct files located in RT1 have been successfully downloaded by T1 (respectively T2).	T1: 30-90KBps; T2: 50-80KBps
Test_11	No problem has been encountered. RT1 is able to download TestCaenF1 from T1 and T2 simultaneously.	RT1: 60-100KBps
Test_12	Not applicable	
Test_13	No problem has been encountered. RT1 has succeeded to download simultaneously TestCaenFa (from T1) and TestCaenFb (from T2).	RT1: 30-50KBps from T1 and 30-40KBps from T2

Table 2: Allow Same IP & PCP Disabled

[5.2.](#) Forbid Same IP Address & PCP Disabled

The following table summarizes the results of the tests when "bt.allow_same_ip == FALSE" in all involved BitTorrent clients and PCP is disabled.

Index	Results	Downloading Speed
Test_1	No problems have been experienced	
Test_2	Both T1 and T2 are able to upload distinct torrent files using the same tracker and the same server.	

Test_3	Only one machine can upload a torrent file referring to the same file. The server ensures that only one single torrent file corresponding to the same file is listed in its base.	
Test_4	Three scenarios have been tested: (1) T1 downloads TestCaenFb but T2 does not download any file from T1 (2) T2 downloads TestCaenFa but T1 does not download any file from T2 (3) T1 downloads TestCaenFb and T2 downloads TestCaenFa in the same time. For all these scenarios, mutual downloading between T1 and T2 is not observed.	
Test_5	No mutual downloading between T1 and T2 was observed.	
Test_6	Both T1 and T2 are able to download distinct files from the BitTorrent infrastructure.	T1: 50-110KBps T2: 60-80KBps

Test_7	Both T1 and T2 are able to download the same file located in several seeders. But for each file it is sending (here TestCaenFRT1) RT1 can allow no more than one unique connection to the same address IP. This is the same behavior for RT2. Mutual downloading between T1 and T2 is not observed.	T1 :100-120KBps, After T1 finished, T2 started 100-120KBps
Test_8	Both T1 and T2 are able to download the	T1:

	file but only one single connection is accepted by RT1 at the same time. This is because for each file it is sending (here TestCaenFRTa) RT1 can allow no more than one unique connection to the same address IP. The result is that, once T1 (or T2) has begun to download TestCaenFRTa, the other terminal (T2 or respectively T1) cannot get any portion of TestCaenFRTa directly from RT1 till the other (T1 or respectively T2) has completed the downloading of TestCaenFRTa. Mutual downloading between T1 and T2 is not observed.	70-100KBps
Test_9	The test has succeeded. While T1 has been downloading TestCaenFRT1 from RT1, T2 could download TestCaenFRTa from RT1 and in addition it can get portions of TestCaenFRTa already downloaded by T1.	T1: 50-70KBps T2: 40-50KBps
Test_10	No problem has been encountered. Distinct files located in RT1 have been successfully downloaded by T1 (respectively T2).	T1: 50-70KBps T2: 40-60KBps
Test_11	Both T1 and T2 are able to upload the file, but only one connection is accepted by RT1 at the same time. The test failed because, once RT1 has begun to download portions of TestCaenF1 from T1 (respectively T2) it cannot accept additional connection with T2 for the same file.	RT1: 20-40KBps from T1

Test_12	The test succeeded. Once T1 has completed its downloading from RT1, T2 has been able automatically to connect	T2: 80-100KBps
---------	---	-------------------

	to RT1 for receiving the same file.	
Test_13	No problem has been encountered. RT1 has succeeded to download simultaneously TestCaenFa (from T1) and TestCaenFb (from T2).	RT1: 30-50KBps from T1 and 30-50KBps from T2

Table 3: Forbid Same IP & PCP Disabled

5.3. Allow Same IP Address & PCP Enabled

The following table summarizes the results of the tests when "bt.allow_same_ip == TRUE" in all involved BitTorrent clients and PCP is enabled.

Index	Results	Downloading Speed
Test_1	No problems have been experienced	
Test_2	Both T1 and T2 are able to upload distinct torrent files using the same tracker and the same server.	
Test_3	Only one machine can upload a torrent file referring to the same file. The server ensures that only one single torrent file corresponding to the same file is listed in its base	
Test_4	Three scenarios have been tested: (1) T1 downloads TestCaenFb but T2 does not download any file from T1 (2) T2 downloads TestCaenFa but T1 does not download any file from T2 (3) T1 downloads TestCaenFb and T2 downloads TestCaenFa in the same time. For all these scenarios, no problems have been encountered. The downloading operations have succeeded.	(1)T1: 1.4-1.5MBps (2)T2: 1.4-1.5MBps (3)T1 and T2: 600-800KBps

Test_5	The mutual downloading operations have succeeded	T1/T2: 1.4-1.5MBps
Test_6	Both T1 and T2 are able to download distinct files from the BitTorrent infrastructure.	T1: 100-110KBps T2: 60-80KBps
Test_7	Both T1 and T2 are able to download the same file located in several seeders. Mutual downloading by T1 of portions of TestCaenFRT1 already downloaded by T2 (and vice versa) has been observed.	T1 and T2: normal speed is 90-140KBps (the highest is 800KBps), between T1 and T2, the normal speed is 50-70KBps (the highest is 700KBps)
Test_8	Both T1 and T2 are able to download TestCaenFRTa from RT1 simultaneously. Mutual downloading by T1 of portions of TestCaenFRTa already downloaded by T2 (and vice versa) has been observed.	T1 and T2: normal speed is 80-110KBps (the highest is 700KBps), between T1 and T2, the normal speed is 40-50KBps (the highest is 600KBps)
Test_9	Not applicable	
Test_10	No problem has been encountered. Distinct files located in RT1 have been successfully downloaded by T1 (respectively T2).	T1: 50-70KBps T2: 40-70KBps
Test_11	No problem has been encountered. RT1 is able to download TestCaenF1 from T1 and T2 simultaneously.	RT1: 60-80KBps
Test_12	Not applicable	

Test_13	No problem has been encountered. RT1 has succeeded to download simultaneously TestCaenFa (from T1) and TestCaenFb (from T2).	RT1: 30-50KBps from T1 and 30-40KBps from T2
---------	--	--

Table 4: Allow Same IP & PCP Enabled

5.4. Forbid Same IP Address & PCP Enabled

The following table summarizes the results of the tests when "bt.allow_same_ip == FALSE" in all involved BitTorrent clients and PCP is enabled.

Index	Results	Downloading Speed
Test_1	No problems have been experienced	
Test_2	Both T1 and T2 are able to upload distinct torrent files using the same tracker and the same server.	
Test_3	Only one machine can upload a torrent file referring to the same file. The server ensures that only one single torrent file corresponding to the same file is listed in its base.	
Test_4	Three scenarios have been tested: (1) T1 downloads TestCaenFb but T2 does not download any file from T1 (2) T2 downloads TestCaenFa but T1 does not download any file from T2 (3) T1 downloads TestCaenFb and T2 downloads TestCaenFa in the same time. For (1) and (2), after several tries, downloading operations have succeeded to	(1)T1: 1.4-1.5MBps (2)T2: 1.4-1.5MBps

	be observed. But for (3), mutual downloading between T1 and T2 is not observed.	
Test_5	The mutual downloading operations have succeeded.	T1/T2: 1.4-1.5MBps

Test_6	Both T1 and T2 are able to download distinct files from the BitTorrent infrastructure.	T1: 100-110KBps T2: 60-70KBps
Test_7	Both T1 and T2 are able to download the same file located in several seeders. But for each file it is sending (here TestCaenFRT1) RT1 can allow no more than one unique connection to the same address IP. This is the same behavior for RT2. Mutual downloading between T1 and T2 is not observed.	T1 :100-120KBps After T1 finished, T2 started 100-120KBps
Test_8	Both T1 and T2 are able to download the file but only one single connection is accepted by RT1 at the same time. This is because for each file it is sending (here TestCaenFRTa) RT1 can allow no more than one unique connection to the same address IP. The result is that, once T1 (or T2) has begun to download TestCaenFRTa, the other terminal (T2 or respectively T1) cannot get any portion of TestCaenFRTa directly from RT1 till the other (T1 or respectively T2) has completed the downloading of TestCaenFRTa. Mutual downloading between T1 and T2 is not observed.	T1: 60-90KBps
Test_9	The test has succeeded. While T1 has been downloading TestCaenFRT1 from RT1,	T1: 50-70KBps

	T2 could download TestCaenFRTa from RT1 and in addition it can get portions of TestCaenFRTa already downloaded by T1.	T2: 40-50KBp
Test_10	No problem has been encountered. Distinct files located in RT1 have been successfully downloaded by T1 (respectively T2).	T1: 50-70KBps T2: 30-50KBps

Test_11	Both T1 and T2 are able to upload the file, but only one connection is accepted by RT1 at the same time. The test failed because, once RT1 has begun to download portions of TestCaenF1 from T1 (respectively T2) it cannot accept additional connection with T2 for the same file.	RT1: 20-40KBps from T1
Test_12	The test succeeded. Once T1 has completed its downloading from RT1, T2 has been able automatically to connect to RT1 for receiving the same file.	T2: 80-100KBps
Test_13	No problem has been encountered. RT1 has succeeded to download simultaneously TestCaenFa (from T1) and TestCaenFb (from T2).	RT1: 30-40KBps from T1 and 40-50KBps from T2

Table 5: Forbid Same IP & PCP Enabled

6. Conclusions

This document describes the main behavior of BitTorrent in an IP shared address environment. The impact of activating port forwarding (here PCP is used) has been also assessed.

Mutual file sharing between hosts sharing the same IP address has been checked. Machines having the same IP address can share files with no alteration compared to current IP architectures only if port forwarding (PCP in our case) is enabled.

Mutual file sharing between hosts behind an IP address sharing function has been also checked. Machines having distinct IP addresses but located behind an address sharing function can share files with no alteration compared to current IP architectures only if port forwarding (PCP in our case) is enabled.

Even if PCP is enabled, two limitations were experienced:

The first limitation occurs when two clients sharing the same IP address want to simultaneously retrieve the SAME file located in a SINGLE remote peer. This limitation is due to the default BitTorrent configuration on the remote peer which does not permit

sending the same file to multiple ports of the same IP address. This limitation is mitigated by the fact that clients sharing the same IP address can exchange portions with each other, provided the clients can find each other through a common tracker, DHT, or Peer Exchange. Even if they can not, we observed that the remote peer would begin serving portions of the file automatically as soon as the other client (sharing the same IP address) finished downloading. This limitation is eliminated if the remote peer is configured with `bt.allow_same_ip == TRUE`.

The second limitation occurs when a client tries to download a file located on several seeders, when those seeders share the same IP address. This is because the clients are enforcing `bt.allow_same_ip` parameter to `FALSE`. The client will only be able to connect to one seeder, among those having the same IP address, to download the file (note that the client can retrieve the file from other seeders having distinct IP addresses). This limitation is eliminated if the local client is configured with `bt.allow_same_ip == TRUE`, which is somewhat likely as those

clients will directly experience better throughput by changing their own configuration.

[7.](#) IANA Considerations

This document raises no IANA considerations.

[8.](#) Security Considerations

This memo does not introduce any security issue.

[9.](#) References

[9.1.](#) Normative References

[I-D.ietf-pcp-base]

Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", [draft-ietf-pcp-base-24](#) (work in progress), March 2012.

[I-D.ietf-pcp-upnp-igd-interworking]

Boucadair, M., Dupont, F., Penno, R., and D. Wing, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD)-Port Control Protocol (PCP) Interworking Function", [draft-ietf-pcp-upnp-igd-interworking-01](#) (work in progress), March 2012.

Boucadair, et al.	Expires November 5, 2012	[Page 20]
-------------------	--------------------------	-----------

Internet-Draft	BitTorrent & PCP	May 2012
----------------	------------------	----------

[9.2.](#) Informative References

[I-D.ietf-behave-lsn-requirements]

Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NATs (CGNs)", [draft-ietf-behave-lsn-requirements-06](#) (work in progress), May 2012.

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", [RFC 6269](#), June 2011.

[RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", [RFC 6346](#), August 2011.

Authors' Addresses

Mohamed Boucadair (editor)
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Tao Zheng
Orange Labs
Beijing
China

Email: tao.zheng@orange.com

NG Tung
Orange Labs
Issy Les Moulineaux
France

Email: paul.ngtung@orange.com

Xiaohing Deng
Orange Labs
Beijing
China

Email: xiaohong.deng@orange.com

Jaqueline Queiroz
Orange Labs
Issy Les Moulineaux
France

Email: jaqueline.queiroz@orange.com