Network Working Group Internet-Draft Intended status: Informational Expires: July 16, 2009 M. Boucadair, Ed. J-L. Grimault P. Levis A. Villefranque France Telecom January 12, 2009

# Behaviour of BitTorrent service in an IP Shared Address Environment draft-boucadair-behave-bittorrent-portrange-02.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of <u>BCP 78</u> and <u>BCP 79</u>.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <a href="http://www.ietf.org/ietf/lid-abstracts.txt">http://www.ietf.org/ietf/lid-abstracts.txt</a>.

The list of Internet-Draft Shadow Directories can be accessed at <a href="http://www.ietf.org/shadow.html">http://www.ietf.org/shadow.html</a>.

This Internet-Draft will expire on July 16, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to <u>BCP 78</u> and the IETF Trust's Legal Provisions Relating to IETF Documents (<u>http://trustee.ietf.org/license-info</u>) 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.

# Abstract

This memo describes the behaviour of BitTorrent service in the context of IP shared addresses. 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.

# Table of Contents

$\underline{1}.  \text{Introduction}  \ldots  \ldots  \ldots  \ldots  \ldots  \ldots  \ldots  \ldots  \ldots  $	· <u>4</u>
<pre>2. BitTorent Overview</pre>	. <u>4</u>
<pre>2.1. BitTorrent at a Glance</pre>	. <u>4</u>
2.2. Software Configuration	. <u>5</u>
<u>2.2.1</u> . BitTorrent Client	. <u>5</u>
2.2.2. BitTorrent Server	. <u>5</u>
2.2.3. BitTorrent Tracker	. <u>6</u>
<u>3</u> . Testbed Overview	. <u>6</u>
<u>3.1</u> . Testbed Description	. <u>6</u>
<u>3.2</u> . Files	· <u>7</u>
4. Description of Tests	. 7
4.1. Connection to Overlav Test Group	. 7
4.2. Upload Test Group	. 7
4.3. Mutual Download Test Group	. 8
4.4. Simultaneous Download Test Group	. <u>9</u>
	10
<u>5</u> . Resulls	• 12
5.1. First configuration: Multiple connections with the	10
Same IP address are enabled	· <u>12</u>
5.2. Second configuration: Multiple connections with the	1.4
	. <u>14</u>
<u>6</u> . Conclusions	. <u>17</u>
7 JANA Considerations	17
$\underline{I}$ . TANA CONSIDERATIONS	• 17
<u>8</u> . Security Considerations	. <u>18</u>
<u>9</u> . Acknowledgements	. <u>18</u>
10. References	. 18
10.1. Normative References	. 18
10.2. Informative References	. 18

Internet-Draft	Port Range BitTorrent	January 2009
Authors' Addresses		<u>19</u>

## **1**. Introduction

Recently, several proposals have been disseminated within IETF to contribute to solve the IP exhaustion problem. These solutions may be grouped into two categories:

(1) Solutions which propose the introduction of a second level of NAT (Network Address Translator), denoted also as Carrier Grade NAT (CG-NAT). This node is located in the Service Provider domain. Private addresses are assigned to end-user CPEs, which still perform their own NAT. The CG-NAT is responsible for translating IP packets issued with private addresses to ones with publicly routable IPv4 addresses (especially when exiting the domain of the Service Provider).

[<u>ID.durand-softwire-dual-stack-lite</u>] is a variant of these solutions where there is only one NAT hosted in the Service Provider's network.

(2) Solutions which avoid the introduction of a NAT in the Service
Provider's network. Examples of these solutions are
[ID.ymbk-aplusp], [ID.boucadair-port-range], [ID.despres-sam] and
[ID.bajko-v6ops-port-restricted-ipaddr-assign]. These solutions
allocate the same IP public address to several customers at the
same time. They also allocate a restricted port range to each
customer so that two customers with the same IP address have two
different port ranges that do not overlap.

Both the above listed categories are based on sharing an IP address between several machines. In this context, the delivery of some services may be impacted, especially those enforcing a restriction based on the source IP address.

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.

Testing activities are conducted using a testbed which is configured according to the solution described in [<u>ID.boucadair-port-range</u>].

## 2. BitTorent Overview

### **<u>2.1</u>**. 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

Boucadair, et al. Expires July 16, 2009 [Page 4]

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.

#### **<u>2.2</u>**. 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

The installed version is 6.1.

#### 2.2.2. BitTorrent Server

The BitTorrent server that has been used is the following:

URL: www.metro-torrent.com

# 2.2.3. BitTorrent Tracker

The BitTorrent tracker that has been used is the following:

URL: www.metro-torrent.com/announce.php

#### 3. Testbed Overview

## 3.1. Testbed Description

The testbed used to conduct the testing activities, described in <u>Section 5</u>, is the same as the one described in Section 4 of [<u>ID.boucadair-port-range</u>].

- o The PRR (Port Range Router) is responsible to implement a portdriven routing so as to be able to route incoming traffic to the appropriate machine among those having the same IP address.
- O CPE-1 and CPE-2 are two CPEs to which the same IP address is assigned. In order to discriminate these CPEs, distinct Port Masks [ID.boucadair-dhc-port-range] are assigned to each of these CPE.
- o T1 (respectively T2) is a machine located in the LAN behind CPE-1
   (respectively CPE-2).

0	RT1	and	RT2	are	remote	machines	reachable	through	Internet.

++ ++	+	-+ +	-+
	Service		
T1   CPE-1	Provider		
	Domain		++
++ ++	I		Remote
193.51.145.206	I		++ Terminal
	++		(RT1)
++ ++	PRR	++ Internet	++
	++		193.51.145.205
T2   CPE-2	I		
++			++
193.51.145.206	I		Remote
	I		++ Terminal
	I		(RT2)
	I		++
	I		193.51.145.208
	+	-+ +	-+

Boucadair, et al. Expires July 16, 2009 [Page 6]

# 3.2. Files

The following table lists the files available in each machine:

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

Available files

## **<u>4</u>**. Description of Tests

This section lists the tests that have been conducted.

## **<u>4.1</u>**. 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.

Boucadair, et al. Expires July 16, 2009 [Page 7]

Internet-Draft

Test	Test Title	Purpose	Description
Index			
Test_2   	Uploading distinct files using the same BitTorrent tracker and server Uploading torrent files referring to the same file	<pre>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 two terminals, having the same public IP address, are able to upload torrent files, which refer to the same file, using the same tracker</pre>	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 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 (resp.

Upload Test Group

# <u>4.3</u>. 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.

Internet-Draft

| Test | Test Title | Purpose | Description | Index | | +----- - - - - - - + - - - -----+ | Test\_4 | Mutual | Check if | Check if T1 can download the | Downloading | two | file uploaded by T2 (ref. | between | terminals | Test\_2) and vice versa. Three | | machines | having | scenarios are to be tested: | sharing the | the same | (1) T1 downloads TestCaenFb | public IP | but T2 does not download any | same IP | address | file from T1, (2) T2 downloads | | address | can | TestCaenFa but T1 does not | download | download any file from T2, (3) | | a file | T1 downloads TestCaenFb and T2 | | from each | downloads TestCaenFa at the | another | same time 

Mutual Download Test Group

## **<u>4.4</u>**. 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 | Test Title | Purpose | Description | Index | | Test\_5 | Downloading | Check if two | Check if distinct files | | distinct | terminals, | available on BitTorrent | | having the | infrastructure may be | | files | same public IP | downloaded by T1 and T2 | | address, are | simultaneously | able to | download | distinct files | | available on T | BitTorrent | infrastructure |

Test_6	Downloading	Check if two	Check if a file
	the same	terminals,	available on several
	file located	having the	seeders may be
	on several	same public IP	downloaded from T1 and
1	seeders	address, are	T2 simultaneously. As
		able to	an example, check if T1
1		download the	and T2 can download the
1		same file	same file located in
		located on	RT1 and RT2 (referred
		several	to as TestCaenFRT1)
1		seeders	
Test_7	Download the	Check it two	Check if I1 and I2 can
Test_7 	Download the     same file	terminals	Check if I1 and I2 can   download the same file
Test_7   	Download the     same file     available on	Check if two terminals having the	Check if I1 and I2 can   download the same file   uploaded by RT1
Test_7     	Download the     same file     available on     a single	Check IT two   terminals     having the   same public IP	Check if I1 and I2 can   download the same file   uploaded by RT1   (referred to as
Test_7     	Download the     same file     available on     a single     machine	Check IT two   terminals   having the   same public IP     address are	Check if I1 and I2 can   download the same file   uploaded by RT1   (referred to as   TestCaenFRTa)
Test_7       	Download the     same file     available on     a single     machine	Check if two terminals having the same public IP address are able to	Check if I1 and I2 can   download the same file   uploaded by RT1   (referred to as   TestCaenFRTa)   concurrently. In case
Test_7         	Download the     same file     available on     a single     machine   	Check if two terminals having the same public IP address are able to download, at	Check if I1 and I2 can download the same file   uploaded by RT1   (referred to as   TestCaenFRTa)   concurrently. In case   the test fails, one of
Test_7           	Download the   same file     available on     a single   machine   	Check IT two terminals having the same public IP address are able to download, at the same time,	Check if I1 and I2 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
Test_7             	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	Check if I1 and I2 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"
Test_7             	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	Check if I1 and I2 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"

Boucadair, et al. Expires July 16, 2009 [Page 10]

Test_8   	Simultaneous downloading from the same seeder	<pre>  Check if it is   not precluded   that a   different file   can be   downloaded by   the waiting   client from   the same   seeder          </pre>	<pre>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</pre>	
             	Downloading	           Check if the	downloading of TestCaenFRT1 and TestCaenFRTa. Check that while T1 is downloading TestCaenFRT1 that does not preclude T2 to concurrently download TestCaenFRTa. Check if T1	
 	distinct files from the same seeder	<pre>  two terminals   two terminals   having the   same public IP   address are   able to   download at   the same time   two distinct   files from the   same seeder  </pre>	Check IT TI(respectively T2) candownload files uploadedby RT1 (referred to asTestCaenRF1 andTestCaenFRTa)concurrently.Particularly, check ifT1 can downloadTestCaenFRT1 and T2 candownload TestCaenFRTasimultaneously	
Test_10             	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	

Boucadair, et al. Expires July 16, 2009 [Page 11]

	Test_11 Test_12	<pre>  Automatic   query to   download the   same file   available on   a single   machine     Download   distinct   files from   two machines   having the   same IP   address</pre>	<pre>  Check if the   In case Test_7 fails,   terminal which   check that the terminal   was waiting   which was waiting can   can finally   finally download the   download the   file once the other   file once the   terminal has finished   other terminal     has finished     Check if   Check if RT1 can   distinct files   download simultaneously   can be   TestCaenFa (from T1)   downloaded by   and TestCaenFb (from   the same   T2)   machine from     seeders having     the same IP  </pre>	
   		address   	seeders having     the same IP     address	   
Т.		· +	T	

#### Simultaneous Download Test Group

#### 5. Results

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).

For the testing activities, two configurations have been tested:

- 1. First Configuration: bt.allow\_same\_ip == TRUE
- 2. Second Configuration: bt.allow\_same\_ip == FALSE

The following sub-sections describe the obtained results for each configuration.

# 5.1. First Configuration: Multiple Connections with the same IP address are enabled

The following table summarises the results of the aforementioned tests as performed using the testbed described in <u>Section 4</u>. Note that bt.allow\_same\_ip is positioned to TRUE.

Boucadair, et al. Expires July 16, 2009 [Page 12]

Internet-Draft

			L
ר   	Test   Identifier	Results	Comments
	Test_1 Test_2	No problems have been experienced Both T1 and T2 are able to upload distinct torrent files using the	None   None
	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	None   
	Test_5	Both T1 and T2 are able to download distinct files from the BitTorrent infrastructure	None
	Test_6	Both T1 and T2 are able to download the same file located in several seeders. No particular problem has been encountered	None
	Test_7	No problem has been encountered. Both T1 and T2 are able to download TestCaenFRTa from RT1 simultaneously. Note that at the same time, mutual downloading by T1 of portions of TestCaenFRTa already downloaded by T2 (and vice versa) have been noticed	None
ĺ	Test_8	Not applicable	None

Boucadair, et al. Expires July 16, 2009 [Page 13]

Test_9	No problem has been encountered.	None	
	Distinct files located in RT1 have		
	been successfully downloaded by T1		
	(respectively T2)		
Test_10	)   No problem has been encountered	None	
Test_11	.   Not applicable	Not	
		applicable	
Test_12	No problem has been encountered.	None	
	RT1 has succeeded to download		
	simultaneously TestCaenFa (from T1)		
	and TestCaenFb (from T2)		
+	+	-+	+

First Configuration Obtained Results

# 5.2. Second Configuration: Multiple Connections with the same IP address are disabled

The following table summarises the results of the aforementioned tests as performed using the testbed described in <u>Section 4</u>. Note that bt.allow\_same\_ip is positioned to FALSE.

+		++
Test   Identifier	Results	Comments   
Test_1 	No problems have been   experienced	None
Test_2       	Both T1 and T2 are   able to upload   distinct torrent   files using the same   tracker and the same   server	None                 
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

Boucadair, et al. Expires July 16, 2009 [Page 14]

Test_4	Three scenarios have	None
	been tested: (1) T1	
	downloads TestCaenFb	
	but T2 does not	
	download any file	
	from T1 (2) T2	
	downloads TestCaenFa	
	but T1 does not	
Ì	download any file	
Ì	from T2 (3) T1	
Ì	downloads TestCaenFb	· 
Ì	and T2 downloads	
Ì	TestCaenFa in the	· 
Ì	same time. For all	· 
Ì	these scenarios, no	
Ì	problems have been	
Ì	encountered. The	
Ì	downloading	
Ì	operations have	
Ì	succeeded	· 
Test_5	Both T1 and T2 are	None
Ì	able to download	· 
Ì	distinct files from	· 
İ	the BitTorrent	
Ì	infrastructure	· 
Test_6	Both T1 and T2 are	When TestCaenFRT1 is used as
Ì	able to download the	example. T1 and T2 are able to
Ì	same file located in	download the same file. But for
Ì	several seeders. No	each file it is sending (here
Ì	particular problem	TestCaenFRT1) RT1 can allow no
1	has been encountered	more than one unique connection
1		to the same address IP. This is
1		the same behaviour for RT2. T1 $\mid$
1		and T2 exchanges the portions of
1	l	the files they stored

Boucadair, et al. Expires July 16, 2009 [Page 15]

Test_7	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. However, that does not preclude the waiting terminal (T2 or T1) to download from the other terminal (T1 or T2) portions of
Test_8	The test 8 has succeeded	TestCaenFRTa already downloaded   from RT1   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
Test_9	No problem have been experienced	None
Test_10	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
Test_11	The test succeeded	Once T1 has completed its   downloading from RT1, T2 has been   able automatically to connect to   RT1 for receiving the portions of   TestCaenFRTa it has not already   got from T2
Test_12	No problem has been encountered. RT1 has succeeded to download simultaneously TestCaenFa (from T1) and TestCaenFb (from T2)	None                       

Boucadair, et al. Expires July 16, 2009 [Page 16]

Results: Multiple connections desiabled

#### 6. Conclusions

This memo describes the main behaviour of BitTorrent service in an IP shared address environment. Particularly, the tests have been carried out on a testbed implementing [ID.boucadair-port-range] solution. The results are, however, valid for all IP shared address based solutions.

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.

Mutual file sharing between hosts having the same IP address has been checked. Indeed, machines having the same IP address can share files with no alteration compared to current IP architectures.

## 7. IANA Considerations

This document raises no IANA considerations.

Boucadair, et al. Expires July 16, 2009 [Page 17]

## 8. Security Considerations

This memo does not introduce any security issue.

#### 9. Acknowledgements

The authors would like to thank Dan WING for suggesting the writing of this memo and for the review of the text.

#### 10. References

#### <u>10.1</u>. Normative References

- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", <u>BCP 9</u>, <u>RFC 2026</u>, October 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", <u>BCP 14</u>, <u>RFC 2119</u>, March 1997.

## <u>10.2</u>. Informative References

[ID.bajko-v6ops-port-restricted-ipaddr-assign]

Bajko, G. and T. Savolainen, "Port Restricted IP Address Assignment", Internet Draft, <u>draft-bajko-v6ops-port-</u> <u>restricted-ipaddr-assign-02.txt</u> (work in progress), November 2008.

[ID.boucadair-dhc-port-range]

Boucadair, M., Grimault, J-L., Levis, P., and A. Villefranque, "DHCP Options for Conveying Port Mask and Port Range Router IP Address", Internet Draft, <u>draft-boucadair-dhc-port-range-01.txt</u> (work in progress), October 2008.

### [ID.boucadair-port-range]

Boucadair, M., Grimault, J-L., Levis, P., and A. Villefranque, "Provider-Provisioned CPE: IPv4 Connectivity Access in the context of IPv4 address exhaustion", Internet Draft, <u>draft-boucadair-port-range-00.txt</u> (work in progress), October 2008.

## [ID.despres-sam]

Despres, R., "Stateless Address Mappings (SAMs) IPv6 & extended IPv4 via local routing domains - possibly multihomed", Internet Draft, <u>draft-despres-sam-01.txt</u> (work in progress), November 2008.

Boucadair, et al. Expires July 16, 2009 [Page 18]

[ID.durand-softwire-dual-stack-lite] Durand , A., Droms, R., Haberman, B., and J. Woodyatt, "Dual-stack lite broadband deployments post IPv4 exhaustion", Internet Draft, <u>draft-durand-softwire-dual-stack-lite-01.txt</u> (work in progress), November 2008. [ID.ymbk-aplusp] Maennel, O., Bush, R., Cittadini, L., and S. Bellovin, "The A+P Approach to the Broadband Provider IPv4 Address Shortage", Internet Draft, <u>draft-ymbk-aplusp-01.txt</u> (work in progress), November 2008. Authors' Addresses Mohamed Boucadair (editor) France Telecom 42 rue des Coutures BP 6243 Caen Cedex 4 14066 France Email: mohamed.boucadair@orange-ftgroup.com Jean-Luc Grimault France Telecom Email: jeanluc.grimault@orange-ftgroup.com Pierre Levis France Telecom Email: pierre.levis@orange-ftgroup.com Alain Villefranque France Telecom Email: alain.villefrangue@orange-ftgroup.com

Boucadair, et al. Expires July 16, 2009 [Page 19]