Network WG Gabor Bajko
Internet Draft Teemu Savolainen
Intended Status: Proposed Standard Nokia
Expires: April 23, 2010 M. Boucadair
P. Levis
France Telecom
October 24, 2009

Port Restricted IP Address Assignment
draft-bajko-pripaddrassign-02

Status of this Memo

Abstract

When IPv6 was designed, the assumption was that the transition from IPv4 to IPv6 will occur way before the exhaustion of the available IPv4 address pool. The unexpected growth of the IPv4 Internet and the hesitation and technical difficulties to deploy IPv6 indicates that the transition may take much longer than originally anticipated.
It is expected that communication using IPv6 addresses will increase during the next few years to come at the expense of communication using IPv4 addresses. The Internet should reach a safety point in the future, where the number of IPv4 public addresses in use at a given time begins decreasing. It is very likely that the IPv4 public address pool currently available at IANA will be exhausted before the internet reaches this safety point. This creates a need to prolong the lifetime of the available IPv4 addresses.

This document defines methods to allocate the same IPv4 address to multiple hosts, with the aim to prolong the availability of public IPv4 addresses, possibly for as long as it takes for IPv6 to take over the demand for IPv4.

Conventions used in this document

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].

Terminology and abbreviations used in this document

Port restricted IPv4 address: an IP address which can only be used in conjunction with the specified ports. Port restriction refers to all known transport protocols (UDP, TCP, SCTP, DCCP).

CGN	Carrier Grade Network Address Translator
CPE	Consumer Premises Equipment, a device that resides between internet service provider's network and consumers' home network.
PRA	Port Restricted IPv4 Address

Table of Content

1. Introduction

There are a number of possible solutions to deal with the problem of transitioning from IPv4 to IPv6; however none of them is a one fits all solution. Different solutions fit different deployment scenarios (see also [ARKK2008]). A tentative comparison is provided in [WING2008].

As complementary solution for the IPv4-IPv6 coexistence period, this document describes a method, using a newly defined DHCPv4 [RFC2131] option that allows servers to assign port restricted IPv4 addresses to clients. By assigning the same IPv4 address to multiple clients, the availability of IPv4 addresses can be, hopefully, prolonged for as long as it takes for IPv6 to take over the demand for IPv4.

The solution described in this document is intended to be used by large ISPs, who as of the date of writing this document, have a large enough IPv4 address pool to be able to allocate one public IPv4 address for each and every client. They expect though that the situation is unsustainable and they will soon not be able to provide every client with a public IPv4 address. Such ISPs have two possibilities to choose from:
- deploy Network Address Translators (NAT), which can be a significant investment for ISPs not having NATs yet. The address space limitations of [RFC1918] may even force these large ISPs to

deploy double NATs, which come with all the harmful behaviour of Carrier Grade NATs (CGN), as described in [MAEN2008]; or
- allocate fragments of the same public IPv4 address directly to multiple clients (which can be CPEs or end hosts), thus avoid the cost of deploying multiple layers of NATs or carrier grade NATs. It is however assumed, that the demand for IPv4 addresses will decrease in the not so distant future, being taken over by IPv6, as the proposal in this draft is not by any means a permanent solution for the IPv4 address exhaustion problem. In fact, some presented deployment scenarios require existence of IPv6 access network.

For ISPs not having NATs yet, a solution not requiring NATs would probably be preferred. For some other ISPs, who already have NATs in place, increasing the capacity of their NATs might be a viable alternative.

In other deployment scenarios, allocation of shared addresses to devices at the edge of the network would result in distribution of NAT functionality to the edges, in some cases even to CPEs [APLUSP].

This document proposes to use new DHCPv4 option to allocate port-restricted IPv4 addresses to the clients. This method is meant to be an IPv4 to IPv6 transition tool, to be only temporarily used during the period when the demand for public IPv4 addresses will exceed the availability of them.

The port restricted IPv4 address option described in this document can be used in various deployment scenarios, some of which are described in [BOUCADAIRARCH], [APLUSP], and [DSLITE].

2. Port Randomization

It is well documented that attackers can perform "blind" attacks against transport protocols. The consequences of these attacks range from throughput-reduction to broken connections or data corruption. These attacks rely on the attacker's ability to guess or know the five-tuple (Protocol, Source Address, Destination Address, Source Port, Destination Port) that identifies the transport protocol instance to be attacked. Most of these attacks can be prevented by randomly selecting the client source port number such that the possibility of an attacker guessing the exact value is reduced. [RANDOMPORT] defines a few algorithms which can select a random port from the available port range. Clients usually have the (1024, 65535) port range at their disposal to select a random, not yet used port.

When an IP address is allocated to multiple clients, the source port range has to be divided between the clients. The smaller the port range, the easier is for an attacker to guess the next port the client is going to use. Therefore, it is imperative to divide the port range between clients sharing the same IP address in such a way that random selection is preserved. This document proposes two different methods for port allocation, which preserves partly or completely the randomness of the source ports:

- o The first mechanism uses a port mask with a bit locator to communicate a range or multiple ranges of ports to a client. Randomness is preserved when the client is able to select a port randomly across all the available port ranges. The algorithms described in [RANDOMPORT] can be used to select a random port from one port range, but implementations may find it difficult to select random ports across port ranges.

- o The second mechanism uses a cryptographic function to preallocate random ports from the entire port range. The key and other input parameters are communicated to the client, which can calculate the ports it can use. The 'side effect' of this mechanism is that the client is forced to use random ports, as a number of random ports allowed to be used by the client are preallocated by the server. When this mode is used, the network equipments in charge of routing the inbound packets towards the clients may require more processing resources.

3. DHCPv4 Option for allocating port restricted public IPv4 address

This section defines new DHCPv4 option, which allows allocation of port restricted IPv4 addresses.

The option layout is depicted below:

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                | Option Code   |    length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Sub-Option 1                              |
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       ...                                     |
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Sub-Option n                              |
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code

Option Code

OPTION-IPv4-PRA (TBD) - 1 byte

Length

An 8-bit field indicating the length of the option excluding the 'Option Code' and the 'Length' fields

Sub-options

A series of DHCPv4 sub-options.

The sub-option layout is depicted below:

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sub-opt Type  |    length     |              DATA             .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                 DATA                                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The sub-option types defined in this document are:

1   port mask

2   random port delegation function

These two options are exclusive with each other (if one is used, the other one is not).

Length

An 8-bit field indicating the length of the sub-option excluding the 'Sub-opt Type' and the 'Length' fields. The value of the length field is 8 when the Sub-opt Type equals 1 and 26 when the sub-opt Type equals 2.

The format of the DATA field when the sub-opt type indicates port mask (value = 1):

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IPv4 address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Port Range Value          |       Port Range Mask         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv4 address
Public IPv4 address

Port Range Value and Port Range Mask
Port Range Value indicates the value of the mask to be applied and Port Range Mask indicates the position of the bits which are used to build the mask.

Section 4 describes how the client derives the allocated port range from the Port Range Value and Port Range Mask values.

The format of the DATA field when the sub-opt type indicates random port delegation function (value = 2):

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IPv4 address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        function               |        starting point         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    number of delegated ports  |        key K               ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
...                                                          ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
...                                                          ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
...                                                          ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
... key K                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IP address
Public IPv4 address

Function
A 16bit field whose value is associated with predefined encryption functions. This specification associates value 1 with the predefined function described in section 5.

Starting Point
A 16bit value used as an input to the specified function

Number of delegated ports
A 16bit value specifying the number of ports delegated to the client for use as source port values

Key K
A 128 bit key used as input to the predefined function for delegated port calculation

4. Port Mask sub-option usage

The port mask sub-option is used to specify one or multiple range of ports pertaining to the given IP address.

Concretely, this option is used to notify a remote DHCP client about the Port Mask to be applied when selecting a port value as a source port. The Port Mask option is used to infer a set of allowed port values. A Port Mask defines a set of ports that all have in common a subset of pre-positioned bits. This ports set is also called Port Range. Two port numbers are said to belong to the same Port Range if and only if, they have the same Port Mask.

A Port Mask contains two fields: Port Range Value and Port Range Mask.

- The 'Port Range Value' field indicates the value of the significant bits of the Port Mask. The 'Port Range Value' is coded as follows:
  - The significant bits are those where "1" values are set in the Port Range Mask. These bits may take a value of "0" or "1 ".
  - All the other bits (non significant ones) are set to "0".

- The 'Port Range Mask' field indicates the position of the significant bits identified by the bit(s) set to "1".

The Port Range Value field indicates the value of the mask to be applied and the Port Range Mask field indicates the position of the bits which are used to build the mask. The "1" values in the Port Range Mask field indicate by their position the significant bits of the Port Range Value (the pattern of the Port Range Value).

For example:

- A Port Range Mask field equal to 1000000000000000 indicates that the first bit (the most significant one) is used as a pattern of the Port Range Value field;

- A Port Range Mask field equal to 0000101000000000 indicates that the 5th and the 7th most significant bits are used as a pattern of the Port Range Value.

The pattern of the Port Range Value is all the fixed bits in the Port Range Value. All the ports the CPE is allowed to use as source ports must have their number in accordance with the pattern.

The Port Range Value is coded as follows:

- The pattern bits of the Port Range Value are those where "1" values are set in the Port Range Mask. These bits may take a value of 0 or 1.
- All the other bits are set to "0".

4.1 Illustration Examples

In each of the three examples below allocation of 2048 ports is done differently. In all examples it is possible for 32 hosts to share the same public IPv4 address. The 4th example illustrates the ability of the procedure to enforce a balanced distribution of port numbers including the well-known-port values.

a) the following Port Range Mask and Port Range Value are conveyed using DHCP to assign a Port Range (from 2048 to 4095) to a given device:

- Port Range Value: 0000100000000000 (2048)
- Port Range Mask: 1111100000000000 (63488)

b) Unlike the previous example, this one illustrates the case where a non Continuous Port Range is assigned to a given customer's device. In this example, the Port Range Value defines 128 Continuous

Port Ranges, each one with a length of 16 port values. Note that the two first Port Ranges are both in the well-known ports span (i.e. 0-1023) but these two ranges are not adjacent.

The following Port Range Mask and Port Range Value are conveyed in DHCP messages:

- Port Range Value : 0000000001010000 (80)
- Port Range Mask : 0000000111110000 (496)

This means that the 128 following Continuous Port Ranges are assigned to the same device:

- from 80 to 95
- from 592 to 607
- ...
- from 65104 to 65119

c) In this example, the Port Range Value defines two Continuous Port Ranges, each one being 1024 ports long:

- Port Range Value : 0000000000000000 (0)
- Port Range Mask : 1111010000000000 (62464)

This means that the two following Continuous Port Ranges are assigned to the same device:

- from 0 to 1023, and
- from 2048 to 3071

d) In this example, 64 continuous Port Ranges are allocated to each CPE (among a set of 4 CPEs sharing the same IPv4 address).

Among the 64 continuous Port Ranges to each CPE, there is always one within the span of the first 1024 well-known port values. Hereafter is given the Port Range Value and Port Range Mask assigned to 2 CPEs (CPE#0 and CPE#3, CPE#1 and CPE#2 being not represented here):

1. CPE#0

- Port Range Value: 0000000000000000 (0)
- Port Range Mask: 0000001100000000 (768)

The CPE#0 has therefore the 64 following Continuous Port Ranges:

- 1st range: 0-255
- ...
- 64th range: 64512-64767

2. CPE#3

   - Port Range Value: 0000001100000000 (768)
   - Port Range Mask: 0000001100000000 (768)

The CPE#2 has therefore the 64 following Continuous Port Ranges:

   - 1st range: 768-1023
   - ...
   - 64th range: 65280-65535

5. Random Port delegation function

Delegating random ports can be achieved by defining a function which takes as input a key 'k' and an integer 'x' within the range (1024, 65535) and produces an output 'y' also within the range (1024, 65535).

The server uses a cryptographical mechanism (described below) to select the random ports for each host. Instead of assigning a range of ports using port mask to the client, the server sends the inputs of a predefined cryptographic mechanism: a key, an initial value, and the number of ports assigned to this host. The client can then calculate the full list of assigned ports itself.

The cryptographical mechanism ensures that the entire 64k port range can be efficiently distributed to multiple hosts in a way that when hosts calculate the ports, the results will never overlap with ports other hosts have calculated (property of permutation), and ports in the reserved range (smaller than 1024) are not used. As the randomization is done crypthographically, an attacker seeing a host

using some port X cannot determine which other ports the host may be using (as the attacker does not know the key).

Calculation of the random port list is done as follows:

The cryptographic mechanism uses an encryption function y = E(K,x) that takes as input a key K (for example, 128 bits) and an integer x (the plaintext) in range (1024, 65535), and produces an output y (the ciphertext), also an integer in range (1024, 65535). This section describes one such encryption function, but others are also possible.

The server will select the key K. When server wants to allocate e.g. 2048 random ports, it selects a starting point 'a' (1024 <= a <= 65536-2048) in a way that the range (a, a+2048) does not overlap with any other active client, and calculates the values E(K,a), E(K,a+1), E(K,a+2), ..., E(K,a+2046), E(K,a+2047). These are the port numbers allocated for this host. Instead of sending the port numbers individually, the server just sends the values 'K', ' a', and '2048'. The client will then repeat the same calculation.

The server SHOULD use different K for each IPv4 address it allocates to make attacks as difficult as possible. This way, learning the K used in IPv4 address IP1 would not help in attacking IPv4 address IP2 that is allocated by the same server to different hosts.

With typical encryption functions (such as AES and DES), the input (plaintext) and output (ciphertext) are blocks of some fixed size; for example, 128 bits for AES, and 64 bits for DES. For port randomization, we need an encryption function whose input and output is an integer in range (1024, 65535).

One possible way to do this is to use the 'Generalized-Feistel Cipher' [CIPHERS] construction by Black and Rogaway, with AES as the underlying round function.

This would look as follows (using pseudo-code):

```
def E(k, x):
    y = Feistel16(k, x)
    if y >= 1024:
          return y
    else:
          return E(k, y)
```

Note that although E(k,x) is recursive, it is guaranteed to terminate. The average number of iterations is just slightly over 1.

Feistel16 is a 16-bit block cipher:

```
def Feistel16(k, x):
    left = x & 0xff
```



```
    right = x >> 8
    for round = 1 to 3:
        temp = left ^ FeistelRound(k, round, right))
        left = right
```

```
            right = temp
        return (right << 8) | left
```

The Feistel round function uses:

```
    def FeistelRound(k, round, x):
        msg[0] = round
        msg[1] = x
        msg[2...15] = 0
        return AES(k, msg)[0]
```

Performance: To generate list of 2048 port numbers, about 6000 calls to AES are required (i.e., encrypting 96 kilobytes). Thus, it will not be a problem for any device that can do, for example, HTTPS (web browsing over SSL/TLS).

Other port generator functions may be predefined in Standards Track documents and allocated a not yet allocated 'function' value within the corresponding sub-option type field.

6. Option Usage

6.1 Client Behaviour

A DHCP client which supports the option defined in this document MUST support both sub-option types.

A DHCP client which supports the extensions defined in this document, SHOULD insert the option OPTION-IPv4-PRA with both sub-option types into DHCPDISCOVER message to explicitly let the server know that it supports port restricted IPv4 addresses.

- o In the port mask sub-option type, the client SHALL set the IPv4 address and Mask Locator fields to all zeros. The client MAY indicate the number of desired ports in Port Range Value-field, or set that to all zeroes.
- o In the random port delegation sub-option type, the client SHALL set the IPv4 address field, key field and starting point field to all zeros. The client MAY indicate in function field which encryption function it prefers, and in the number of delegated ports field the number of ports the client would desire.

When a client, which supports the option defined in this document, receives a DHCPOFFER with the 'yiaddr' (client IP address) field set to 0.0.0.0, it SHOULD check for the presence of OPTION-IPv4-PRA option. If such an option is present, the client MAY send a DHCPREQUEST message and insert the option OPTION-IPv4-PRA with the corresponding sub-option received in the OPTION-IPv4-PRA option of the previous DHCPOFFER. The client MUST NOT include a 'Requested IP Address' DHCP option (code 50) into this DHCPREQUEST.

The client MUST NOT insert the IP address received in OPTION-IPv4-PRA into the 'Requested IP Address' DHCP option (code 50). When the client receives a DHCPACK message with an OPTION-IPv4-PRA option, it MAY start using the specified IP address in conjunction with the source ports specified by the mechanism chosen by DHCP server. The client MUST NOT use the IP address with different source port numbers, as that may result in a conflict, since the same IP address with a different source port group may be assigned to a different client. Furthermore, the client MUST notice the situation where an outgoing IP packet has the same IP address as destination address than the client itself has, but the port number is not belonging to the allocated set. In this case the client MUST detect that the packet is not destined for itself, and it MUST send it forward.

In case the initial port set received by the client from the server is exhausted and the client needs additional ports, it MAY request so by sending a new DHCPDISCOVER message.

In some deployment scenarios the DHCP client may also act as a DHCP server for a network behind it, in which case the host may further split the allocated set for other hosts.

The allocated port-restricted IP address and all the associated parameters are valid until indicated in the IP Address Lease Time Option (option 51).

6.2 Server Behaviour

When a server, which supports the option defined in this document, receives a DHCPDISCOVER message, it SHOULD check the presence of the OPTION-IPv4-PRA option.

If OPTION-IPv4-PRA is not present in DHCPDISCOVER, the server SHOULD allocate full unrestricted public or private [RFC1918] IPv4 address to the client, if available, by generating a DHCPOFFER as described in [RFC2131].

The server SHOULD offer the port restricted IPv4 address when the server has support for the extensions specified in this document and when:

- o DHCP client has included an OPTION-IPv4-PRA option, and server's policy indicates saving unrestricted IPv4 addresses for clients that do not support the extensions defined in this document. The server MUST include only one of the sub-options into the

OPTION-IPv4-PRA option (the one which it uses for port restricted IP address allocation).

- o server receives a DHCPDISCOVER message and server can only offer port restricted IP address to the client

- o server receives a DHCPDISCOVER message from a client without the OPTION-IPv4-PRA, but knows by means outside the scope of this document that the client supports the usage of port-restricted IPv4 addresses (or it is only entitled to be provisioned with such addresses)

When server chooses to offer port restricted IPv4 address for clients with OPTION-IPv4-PRA, it MUST:

- o set the 'yiaddr' (client IP address) field of the DHCPOFFER message to 0.0.0.0
- o choose the port allocation mechanisms, if it is not statically configured
- o select a port restricted IPv4 address to be allocated for the client
- o generate parameters required for the chosen port allocation mechanism

When the server receives a DHCPREQUEST message from the client with an OPTION-IPv4-PRA option field containing the IP address and port allocation mechanism parameters it has previously offered to the client, the server MUST send a DHCPACK, where the 'yiaddr' (client IP address) field is set to 0.0.0.0 and the OPTION-IPv4-PRA option including the IPv4 address and parameters required for the used allocation mechanism.

When the server receives a DHCPREQUEST message from the client with an OPTION-IPv4-PRA option field containing an IPv4 address and port set it has previously not offered to the client, the server MUST send a DHCPNAK to the client.

When the server detects that a client (by eg having a specific hardware address) which has already been allocated with a port restricted IPv4 address, sent another DHCPDISCOVER, it MAY, based on local policy, offer the client with additional port restricted IPv4 address.

If the server is deployed in a cascaded DHCP server scenario, the host MAY both act as a DHCP client for another server and DHCP server for other DHCP clients.

A server SHOULD ensure the client is residing on an access link where usage of port-restricted addresses is not causing problems, before allocating it a port restricted IPv4 address.

The server MUST keep lease times per allocated port sets of the shared IP addresses.

7. Applicability

The multiplexing of IP flows in gateway is based on the port numbers used by transport layer protocols such as TCP, UDP, SCTP, and DCCP.

However, the protocols not containing port numbers need special handling in order to be multiplexed correctly.

7.1 ICMP

Those ICMP messages that embed the IP packet that triggered sending of ICMP message, such as ICMP error, can be multiplexed based on the port number present in the embedded original packet.

ICMP messages not containing embedded packets, like ICMP echo, are TBD.

7.2 6to4

A host utilizing 6to4 [RFC3056] with port restricted IPv4 addresses MUST pick the 16-bit .SLA ID. value for the 6to4 prefix(es) construction from the pool of allocated port values. The multiplexing gateway MUST then multiplex 6to4 traffic based on .SLA ID. value as it would multiplex plain IPv4 traffic based on port values. I.e. for incoming packets the gateway shall look at the destination IPv4 address and the .SLA ID.-field from tunneled IPv6 packet.s destination IPv6 address, and then select the right route as it would have picked the port number from a transport layer header.

7.3 Protocols not supported by multiplexing gateway

The case where port range router is not able to multiplex a protocol is similar to a case where middle box, such as firewall or NAT, blocks traffic it is not able or willing to pass trough. The application is recommended to fallback to UDP encapsulation often used for NAT traversal, for which gateway is able to perform multiplexing.

8. IANA considerations

This document defines new DHCPv4 option as described in section 3: Port Restricted IP Address Option for DHCPv4 (OPTION-IPv4-PRA) TBD.

9. Security considerations

The solution is generally vulnerable to DoS when used in shared medium or when access network authentication is not a prerequisite to IP address assignment. The solution SHOULD only be used on point-to-point links, tunnels, and/or in environments where authentication at link layer is performed before IP address assignment, and not shared medium.

The cryptographically random port delegation mechanism is vulnerable for blind attacks initiated by hosts located in the same administrative domain, served by the same DHCP server, and that are sharing the same public IPv4 address, and therefore have knowledge of the cryptographic key used for that particular public IPv4 address.

10. Normative References

[RFC2119] Bradner, S., .Key words for use in RFCs to Indicate Requirement Levels., March 1997

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC2131, March 1997

[RFC3056] Carpenter, B., Moore, K., .Connection of IPv6 Domains via IPv4 Clouds., February 2001

11. Informative References

[ARKK2008] Arkko, J., Townsley, M., "IPv4 Run-Out and IPv4-IPv6 Co-Existence Scenarios", September 2008, draft-arkko-townsley-coexistence-03

[WING2008] Wing, D., Ward, D., Durand, A., "A Comparison of Proposals to Replace NAT-PT", September 2008, draft-wing-nat-pt-replacement-comparison

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., J. de Groot, G., Lear, E., "Address Allocation for Private Internets", RFC1918, February 1996

[MAEN2008] Maennel, O., Bush, R., Cittadini, L., Bellovin, S., "A Better Approach than Carrier-Grade-NAT", 2008, Technical Report CUCS-041-08

[RANDOMPORT] Larsen, M., Gont, F., .Port Randomization., August 2008, draft-ietf-tsvwg-port-randomization-04

[CIPHERS] John Black and Phillip Rogaway: .Ciphers with Arbitrary Finite Domains., Topics in Cryptology - CT-RSA 2002, Lecture Notes in Computer Science vol. 2271, 2002

[DSLITE] A. Durand et al .Dual-stack lite broadband deployments post IPv4 exhaustion., November 2008, draft-durand-softwire-dual-stack-lite

[BOUCADAIR] Boucadair, M, Ed., Grimault, J-L., Levis, P., Villefranque, A., .DHCP Options for Conveying Port Mask

and Port Range Router IP Address., October 2008, draft-boucadair-dhc-port-range

[BOUCADAIRARCH] Boucadair, M., Ed., Levis, P., Bajko, G., Savolainen, T., .IPv4 Connectivity Access in the Context of IPv4 Address Exhaustion., January 2009, draft-boucadair-port-range

[APLUSP] Maennel, O., Bush, R., Cittadini, L., Bellovin, S., "The A+P Approach to the IPv4 Address Shortage", January 2009, draft-ymbk-aplusp

12. Contributors

The port range allocation using Port Range Value / Port Range Mask comes from [BOUCADAIR], authored by Mohamed Boucadair, Jean Luc

Grimault and Pierre Lewis.

The encryption function from section 5 was provided by Pasi Eronen.

The text on 6to4 handling was proposed by Dave Thaler.

The rest of the document was written and edited by Gabor Bajko and Teemu Savolainen.

The authors would also like to thank Lars Eggert, Olaf Maenel, Randy Bush, Alain Durand, Jean-Luc Grimault, Alain Villefranque for their valuable comments.

13. Authors' Addresses

Gabor Bajko
gabor(dot)bajko(at)nokia(dot)com

Teemu Savolainen
Nokia
Hermiankatu 12 D
FI-33720 TAMPERE
Finland

Email: teemu.savolainen@nokia.com

Mohamed Boucadair
France Telecom
42 rue des Coutures
BP 6243
Caen Cedex 4 14066
France

Email: mohamed.boucadair@orange-ftgroup.com

Pierre Levis
France Telecom
42 rue des Coutures
BP 6243
Caen Cedex 4 14066

France

Email: pierre.levis@orange-ftgroup.com