Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 30, 2010

R. Bush, Ed.
Internet Initiative Japan

October 27, 2009

The A+P Approach to the IPv4 Address Shortage
draft-ymbk-aplusp-05

Status of this Memo

Copyright Notice

Abstract

We are facing the exhaustion of the IANA IPv4 free IP address pool.

Unfortunately, IPv6 is not yet deployed widely enough to fully replace IPv4, and it is unrealistic to expect that this is going to change before we run out of IPv4 addresses. Letting hosts seamlessly communicate in an IPv4-world without assigning a unique globally routable IPv4 address to each of them is a challenging problem.

This draft discusses the possibility of address sharing by treating some of the port number bits as part of an extended IPv4 address (Address plus Port, or A+P). Instead of assigning a single IPv4 address to a customer device, we propose to extended the address by "stealing" bits from the port number in the TCP/UDP header, leaving the applications a reduced range of ports. This means assigning the same IPv4 address to multiple clients (e.g., CPE, mobile phones), each with its assigned port-range. In the face of IPv4 address exhaustion, the need for addresses is stronger than the need to be able to address thousands of applications on a single host. If address translation is needed, the end-user should be in control of the translation process - not some smart boxes in the core.

Requirements Language

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].

Table of Contents

1. Introduction

This document describes a technique to deal with the imminent IPv4 address space exhaustion. Many large Internet Service Providers (ISPs) face the problem that their networks' customer edges are so large that it will soon not be possible to provide each customer with a unique IPv4 address. Therefore these ISPs have to devise something more ingenious. Although undesirable, address sharing, a la NAT, is inevitable.

To allow end-to-end connectivity between IPv4 speaking applications we propose to "steal" some bits from the UDP/TCP header and use them to extend addressing of devices. Assuming we could limit the applications' port addressing to 8 (or 4) bits, we can increase the effective size of an IPv4 address by 8 (or 12) additional bits. In this scenario, 128 (or 4096) customers could be multiplexed on the same IPv4 address, while allowing them a fixed range of 512 (or 16) ports. Customers that require larger port-ranges could dynamically request additional blocks, depending on their contract. We call this "extended addressing" or "A+P" (Address plus Port) addressing. The main advantage of A+P is that it preserves the Internet "end-to-end" paradigm by not translating (at least some ports of) an IP address. With NAT in the core of the network, this end-to-end connectivity is broken. As long as the customer chooses to do this on his/her premises this is a choice that he/she takes, however this is not an option in face of the looming IPv4 address exhaustion, where so called Carrier Grade NATs (CGNs) might be deployed within the providers network - beyond control of the customer. CGNs come with different names and in different flavors, such as NAT444, Large Scale NATs (LSNs) or Address Family Transition Routers (AFTR).

1.1. Why Carrier Grade NATs are Harmful

Various forms of NATs will be installed at various levels and places in the IPv4-Internet to achieve address compression. This document argues for mechanisms where this happens as close to the edge as possible, thereby minimizing damage to the End to End Principle. End-customers will not be locked into a walled-garden without any control over the translation. It is is essential to create mechanisms to "bypass" NATs in the core, and keep the control at the end-user:

"Carrier grade" is a euphemism for centralized. More semantics move to the core of the network. This is bad in and of itself. Net-heads call it "telco-think" because it is the telco model of smarts in the core as opposed to the Internet model of a simple, just-forward-packets core, with smart edges. It also places the provider in the position, where the user is trapped behind unchangeable application

policies, and has the danger of invoking lawyers when users wish to deploy new applications needing Application Level Gateways (ALGs). This is the opposite of the "end-to-end" model of the Internet.

With the smarts at the edges, one can easily field new protocols between consenting end-points by merely tweaking the NATs at the corresponding Customer Premises Equipment (CPE), even adding application layer gateways if they are needed.

Today's NATs are typically mitigated by ALGs over which the customer has control, e.g. port forwarding or UPnP/NAT-PMP. However, this is not expected to work with CGNs. CGN proposals - other than DS-Lite [I-D.ietf-softwire-dual-stack-lite] with A+P - admit that it is not expected that applications that require specific port assignment or port mapping from the NAT box will keep working. This is the ultimate horror the NAT-haters fear, and, in this case, they are not all that wrong.

We believe this CGN approach is not an option and that the end-user must have the ability to control their own ALGs. With CGN, if a user wishes to deploy a new application, they must talk to the providers' lawyers or run new disruptive technology over HTTP; we can pick our poison. And if the NAT is not where the customer can directly control it, i.e., it is anywhere in the provider's network, then the provider controls what the user can control, i.e. it is not really under user control. We do not wish to deal with the case where the provider has to decide whether to allow Skype v42 when they themselves provide a competing VoIP product.

Another issue with CGN is scalability. ISPs face a tension between the placement of CGNs within their network to aggregate as much as possible, when too much aggregation creates a massive state problem. CGNs also present a single point of failure. And having a back-up CGN has the state transfer problem as well as exposure to network partition and dual-device failure. When you start talking about 'high reliability/availability, you have already lost the game. The internet is about building a reliable network using unreliable devices.

To reduce the state, NAT placement ends up as CGNs somewhere closer to the edge. It is not clear how a CGN should maintain per-session state in a scalable manner. State for improperly terminated sessions could remain stale for some time. The CGN hence trades scalability for the amount of state that needs to be kept, which makes optimally placing a CGN a hard engineering problem.

Furthermore, with CGN, tracing hackers, spammers and other criminals will be impossible, unless all the connection based mapping

information is recorded and stored. This would not only cause concern for law enforcement services, but also for privacy advocates.

2. Design Constraints and Assumptions

The problem of address space shortage is first felt by providers with a very large end-user customer base, such as broadband providers and mobile-service providers. Though the cases and requirements are slightly different, they share many commonalities. In the following we will develop a set of overall design constraints.

2.1. Design constraints

We regard several constraints as important for our design:

1) End-to-End is under customer control: Customers shall have the ability to deploy new application protocols at will. IPv4 address shortage should not be a license to break the Internet's end-to-end paradigm.

2) End-to-End transparency through multiple intermediate devices: Multiple gateways should be able to operate in sequence along one data path without interfering with each other.

3) Backward compatibility: Approaches should be transparent to unaware users. Devices or existing applications should be able to work without modification. Emergence of new applications should not be limited.

4) Incrementally deployable: The provider should not be forced to replace unaffected core devices or replace customer premises equipment (CPE). In particular, the provider should be able to change only CPE where they wish to deploy A+P. And customers should be able to acquire A+P aware CPE at will.

5) Highly-scalable and minimal state core: Minimal state should be kept inside the ISP's network. If the operator is rolling out A+P incrementally, it is understood there may be state in the core in the non-A+P part of such a roll-out.

6) Efficiency vs. complexity: Operators should have the flexibility to trade off port multiplexing efficiency and scalability and end-to-end transparency.

7) Automatic configuration/administration: There should be no need for customers to call the ISP and tell them that they are operating their own A+P-gateway devices. Customers/ mobile phone users should not be expected to look-up assigned ports manually on websites and then configure them on devices or applications.

8) "Double-NAT" should be avoided: Based on Constraint 2 multiple gateway devices might be present in a path, and once one has done some translation, those packets should not be re-translated.

9) Legal traceability: ISPs must be able to provide the identity of a customer from the knowledge of the IPv4 public address and the port. This should have as low an impact as is reasonable on storage by the ISP. We assume that NATs on customer premises do not pose much of a problem, while provider NATs need to keep additional logs.

10) IPv6 deployment should be encouraged. NAT444 strongly biases the users to the deployment of RFC 1918 addressing. A+P should not. While we acknowledge that A+P might be used in an IPv4-only environment (e.g., [I-D.boucadair-port-range]) we strongly believe that IPv6 is the best long-term approach, and that A+P should be considered only as an intermediate hack towards an IPv6-only world. We therefore prefer to assume in Constraint 10 that the ISP has migrated to a dual-stack core and A+P can use IPv6 as a transport inside the network. This ensures that A+P will not be a hindrance to the introduction of IPv6.

Constraints 2 and 8 are important: while many techniques have been deployed to allow applications to work through a NAT, traversing cascaded NATs is crucial if NATs are being deployed in the core of a provider network.

2.2. Terminology

The A+P architecture can be split into three distinct functions: encaps/decaps, NAT, and signaling.

Encaps/decaps function: is used to forward port-restricted A+P-packets over intermediate legacy devices. The encapsulation function takes an IPv4 packet, looks up the IP and TCP/UDP headers, and puts the packet into the appropriate tunnel. The state needed to perform this action is comparable to a forwarding table. The decapsulation device SHOULD check if the source address and port of packets coming out of the tunnel are legitimate (e.g., see [BCP38]). Based on the

result of such a check, the packet MAY be forwarded untranslated, it MAY be discarded or MAY be NATed. In this draft we refer to a device that provides this encaps/decaps functionality as Port-Range-Router (PRR).

Network Address Translation (NAT) function: is used to connect legacy end-hosts. Unless upgraded, end-hosts or end-systems are not aware of A+P restrictions and therefore assume a full IP address. The NAT function performs any address or port translation, including application-level-gateways (ALGs). The state that has to be kept to implement this function is the mapping for which external addresses and ports have been mapped to which internal addresses and ports, just as in CPE NATs today. A subtle, but very important, difference should be noted here: the customer has control over the NATing process or might choose to "bypass" the NAT. If this is done, we call the NAT a large scale NAT (LSN). However, if the NAT that does NOT allow the customer to control the translation process, we refer to as a CGN.

Signaling function: is used in order to allow A+P-aware devices get to know which ports are assigned to be passed through untranslated and what will happen to packets outside the assigned port-range (e.g., could be NATed or discarded). Signaling may also be used to learn the encapsulation method and any endpoint information needed. In addition, the signaling function may be used to dynamically increase/decrease the requested port-range.

A+P address realm: a public routable IPv4 address that is port restricted (A+P). Forwarding of packets is done based on the IPv4 address and the TCP/UDP port numbers. When this draft talks about "A+P packets" it is assumed that those packets pass untranslated.

Private address realm: IPv4 addresses that are not globally routed. They may be taken from the [RFC1918] range. However, this draft does not make such an assumption. We regard as private address space any IPv4 address, which needs to be translated in order to gain global connectivity, irrespective of whether it falls in [RFC1918] space or not.

3. Overview of the A+P Solution

The core architectural elements of the A+P solution are three separated and independent functions: the NAT function, the encaps/decaps function, and the signaling function. The NAT function is similar to a NAT as we know it today: it performs a translation between two different address realms. When the external realm is public IPv4 address space, we assume that the translation is many-to-

one, in order to multiplex many customers on a single public IPv4 address. The only difference with a traditional NAT (Figure 1) is that the translator might only be able to use a restricted range of ports when mapping multiple internal addresses onto an external one, e.g., the external address realm might be port-restricted.

```
                  "internal-side"          "external-side"
                                 +------+
                    internal     |  N   |     external
                    address  <---|  A   |---> address
                     realm       |  T   |      realm
                                 +------+
```

Traditional NAT

Figure 1

The encaps/decaps function, on the other hand, is the ability to establish a tunnel with another end-point providing the same function. This implies some form of signaling to establish a tunnel. Such signaling can be viewed as integrated with DHCP or as a separate service. Section 3.1 discusses the constraints of this signaling function. The tunnel can be an IPv6 or IPv4encapsulation, a layer-2 tunnel, or some other form of softwire. Note that the presence of a tunnel allows unmodified, naive, or even legacy devices between the two endpoints.

Two or more devices which provide the encaps/decaps function and are linked by tunnels to form an A+P subsystem. The function of each gateway is to encapsulate and decapsulate respectively. Figure 2 depicts the simplest possible A+P subsystem, that is, two devices providing the encaps/decaps function.

```
                    +-------------------------------------+
  port-restricted   | +-----------+  tunnel  +-----------+ |   external
   address realm --|-| gateway  |==========| gateway  |-|-- address
                    | +-----------+          +-----------+ |    realm
                    +-------------------------------------+
                                A+P subsystem
```

A simple A+P subsystem

Figure 2

Within an A+P subsystem, the external address realm is extended by "stealing" bits from the port number. Each device is assigned one address from the external realm and a range of port numbers. Hence, devices which are part of an A+P subsystem can communicate with the external address without the need for address translation (i.e., preserving end-to-end packet integrity): an A+P packet originated from within the A+P subsystem can be simply forwarded over tunnels up to the endpoint, where it gets decapsulated and routed in the external realm.

3.1. Signaling

The following information needs to be available on all the gateways in the A+P subsystem. It is expected that there will be a signaling protocol such as [I-D.bajko-pripaddrassign], [I-D.boucadair-dhcpv6-shared-address-option], or [I-D.boucadair-pppext-portrange-option]. The information that needs to be shared is the following:

- a set of public IPv4 addresses,
- for each IPv4 address a starting point for the allocated port-range,
- number of delegated ports,
- optional key that enables partial or full preservation of entropy in port randomization - see [I-D.bajko-pripaddrassign],
- lifetime for each IPv4 address and allocated port-set,
- the tunneling technology to be used (e.g., "IPv6-encapsulation")
- addresses of the tunnel endpoints (e.g., IPv6 address of tunnel endpoints)
- whether or not NAT function is provided by the gateway
- a device identification number and some authentication mechanisms
- a version number and some reserved bits for future use.

Note that the functions of encapsulation and decapsulation have been separated from the NAT function. However, to accommodate legacy hosts, NATing is likely to be provided at some point in the path; therefore the availability or absence of NATing MUST be communicated in signaling, as A+P is agnostic about NAT placement.

The port-ranges can be allocated in two different ways:

- o If applications or end-hosts behind the CPE are not UPnPv2/NAT-PMP aware, then the CPE SHOULD request ports via mechanisms, e.g. as described in [I-D.bajko-pripaddrassign] and [I-D.boucadair-pppext-portrange-option]. Note that different port-ranges can have different lifetimes, and the CPE is not entitled to use them after they expire - unless it refreshes those ranges. It is up to the ISP to put mechanisms in place, that determine what percentage of already allocated port-ranges should be exhausted before a CPE may requests additional ranges, how often the CPE can request additional ranges, and so on. (To prevent Denial of Service attacks.)

- o If applications behind the CPE are UPnPv2/NAT-PMP aware additional ports MAY be requested through that mechanism. In this case the CPE should forward those requests to the LSN and the LSN should reply reporting if the requested ports are available or not (and if they are not available some alternatives should be offered). Here again, to prevent potential denial of service attacks, mechanism should be in place to prevent UPnPv2/NAT-PMP packet storms and fast port allocation.

Whatever signaling mechanism is used inside the tunnels, DHCP or IPCP based, synchronization between signaling server and PRR must be established in both directions. For example, if we use DHCP as signaling mechanism, the PRR must communicate to DHCP server at least its IP range. The DHCP server then starts to allocate IPs and port-ranges to CPEs and communicates back to the PRR which IP and port range have been allocated to which CPE, so the PRR knows to which tunnel redirect incoming traffic. In addition, DHCP MUST also communicate lifetimes of port-ranges assigned to CPE via the PRR.

If UPnPv2/NAT-PMP is used as dynamic port allocation mechanism, the PRR must also communicate to the DHCP (or IPCP) server to avoid those ports. The PRR must somehow (DHCP or IPCP options) communicate back to CPE that allocation of ports was successful, so CPE adds those ports to existing port-ranges.

3.2. Address realm

Each gateway within the A+P subsystem manages a certain portion of A+P address space, that is, a portion of IPv4 space which is extended by borrowing bits from the port number. This address space may be a single, port-restricted IPv4 address. The gateway MAY use its managed A+P address space for several purposes:

- o Allocation of a sub-portion of the A+P address space to other authenticated A+P gateways in the A+P subsystem (referred to as delegation). We call the allocated sub-portion delegated address space.

- o Exchange of (untranslated) packets with the external address realm. For this to work, such packets MUST use source address and port belonging to the non-delegated address space.

If the gateway is also capable of performing the NAT function, it MAY translate packets arriving on an internal interface which are outside of its managed A+P address space into non-delegated address space.

Hence, a provider may have 'islands' of A+P as they slowly deploy over time. The provider does not have to replace CPE until they want to provide the A+P function to an island of users or even to one particular user in a sea of non-A+P users.

An A+P gateway ("A"), accepts incoming connections from other A+P gateways ("B"). Upon connection establishment (provided appropriate authentication), B would "ask" A for delegation of an A+P address. In turn, A will inform B about its public IPv4 address, and will delegate a portion of its port-range to B. In addition, A will also negotiate the encaps/decaps function with B (e.g., let B know the address of the decaps device/other-end-point of the tunnel).

This could be implemented for example via a NAT-PMP or DHCP-like solution. In general the following rule applies: A sub-portion of the managed A+P address space is delegated as long as devices below ask for it, otherwise private IPv4 is provided to support legacy hosts.

```
           private    +-----+          +-----+     public
           address ---|  B  |==========|  A  |---  Internet
            realm     +-----+          +-----+

                      Address space realm of A:
                      public IPv4 address = 12.0.0.1
                      port range = 0-65535

                      Address space realm of B:
                      public IPv4 address = 12.0.0.1
                      port range = 2560-3071
```

Figure 3

Figure 3 illustrates a sample configuration. Note that A might actually consist of three different devices: one that handles signaling requests from B; one device that performs encapsulation and decapsulation; and, if provided, one device that performs NATing function (e.g., LSN). Packet forwarding is assumed to be as follows: In the "out-bound" case, a packet arrives from the private address realm to B. As stated above, B has two options: it can either apply or not apply the NAT function. The decision depends upon the specific configuration and/or the capabilities of A and B. Note that NAT functionality is required to support legacy hosts, however, this can be done at either of the two devices A or B. The term NAT refers to translating the packet into the managed A+P address (B has address 12.0.0.1 and ports 2560-3071 in the example above). We then have two options:

1) B NATs the packet. The translated packet is then tunneled to A. A recognizes that the packet has already been translated, because the source address and port match the delegated space. A decapsulates the packet and releases it in the public Internet.

2) B does not NAT the packet. The untranslated packet is then tunneled to A. A recognizes that the packet has not been translated, so A forwards the packet to a co-located NATing device, which translates the packet and routes it in the public Internet. This device, e.g., an LSN, has to store the mapping between the source port used to NAT and the tunnel where the packet came from, in order to correctly route the reply. Note that A cannot use a port number from the range that has been delegated to B. As a consequence A has to assign a part of its non-delegated address space to the NATing function.

"Inbound" packets are handled in the following way: a packet from the public realm arrives at A. A analyzes the destination port number to

understand whether the packet needs to be NATed or not.

1) If the destination port number belongs to the range that A delegated to B, then A tunnels the packet to B. B NATs the packet using its stored mapping and forwards the translated packet to the private domain.

2) If the destination port number is from the address space of the LSN, then A passes the packet on to the co-located LSN which uses its stored mapping to NAT the packet into the private address realm of B. The appropriate tunnel is stored as well in the mapping of the initial NAT. The LSN then encapsulates the packet to B, which decapsulates it and normally routes it within its private realm.

3) Finally, if the destination port number neither falls in a delegated range, nor into the address range of the LSN, A discards the packet. If the packet is passed to the LSN, but no mapping can be found, the LSN discards the packet.

3.3. Reasons for allowing multiple A+P gateways

Since each device in an A+P subsystem provides the encaps/decaps function, new devices can establish tunnels and become in turn part of an A+P subsystem. As noted above, being part of an A+P subsystem implies the capability of talking to the external address realm without any translation. In particular, as described in the previous section, a device X in an A+P subsystem can be reached from the external domain by simply using the public IPv4 address and a port which has been delegated to X. Figure 4 shows an example where three devices are connected in a chain. In other words, A+P signaling can be used to extend end-to-end connectivity to the devices which are in an A+P subsystem. This allows A+P-aware applications (or OSes) running on end hosts to enter an A+P subsystem and exploit untranslated connectivity.

There are two modes for end-hosts to gain fine-grained control of end-to-end connectivity. The first is where actual end-hosts perform the NAT function and the encaps/decaps function which is required to join the A+P subsystem. This option works in a similar way to the NAT-in-the-host trick employed by virtualization software such as VMware, where the guest operating system is connected via a NAT to the host operating system. The second mode is applications which autonomously ask for an A+P address and use it to join the A+P subsystem. This capability is necessary for some applications that require end-to-end connectivity (e.g., applications that need to be contacted from outside).

```
            +---------+      +---------+      +---------+
  internal  | gateway |      | gateway |      | gateway |  external
  realm   --|    1    |======|    2    |======|    3    |-- realm
            +---------+      +---------+      +---------+
```

An A+P subsystem with multiple devices

Figure 4

Whatever the reasons might be, the Internet was built on a paradigm that end-to-end connectivity is important. A+P makes this still possible in a time where address shortage forces ISPs to use NATs at various levels. In such sense, A+P can be regarded as a way to bypass NATs.

```
           +---+          (customer2)
           |A+P|-.         +---+
           +---+  \     NAT|A+P|-.
                   \       +---+ |
                    \            |       forward if in-range
           +---+     \+---+    +---+    /
           |A+P|------|A+P|----|A+P|----
           +---+     /+---+    +---+    \
                    /                    NAT if necessary
                   / (cust1)   (prov.    (e.g., provider NAT)
           +---+  /            router)
           |A+P|-'
           +---+
```

A complex A+P subsystem

Figure 5

Figure 5 depicts a complex scenario, where the A+P subsystem is composed by multiple devices organized in a hierarchy. Each A+P gateway decapsulates the packet and then re-encapsulates it again to the next tunnel.

A packet can either be NATed when it enters the A+P subsystem, or at intermediate devices, or when it exits the A+P subsystem. This could be for example a gateway installed within the provider's network, together with a LSN. Then each customer operates its own CPE. However, behind the CPE applications might also be A+P-aware and run their own A+P-gateways, which enables them to have end-to-end connectivity.

One limitation applies, if "delayed translation" is used (e.g., translation at the LSN instead of the CPE). If devices using "delayed translation" want to talk to each other they SHOULD use A+P addresses or out-of-band addressing.

4. Deployment Scenarios

4.1. A+P for Broadband Providers

Large broadband providers do not have enough IPv4 address space to provide every customer with a single IP. The natural solution is sharing a single IP address among many customers. Multiplexing customers is usually accomplished by allocating different port numbers to different customers somewhere within the network of the provider.

In this document we use the following terms and assumptions:

1. Customer Premises Equipment (CPE), i.e. cable/DSL modem.

2. Provider Edge Router (PE), AKA customer aggregation router

3. Port Range Router (PRR), edge behind which A+P addresses are used.

4. Provider Border Router (BR), providers edge to other providers

5. Network Core Routers (Core), provider routers which are not at the edge.

It is expected that, when the provider wishes to enable A+P for a customer or a range of customers, the CPE can be upgraded or replaced to support A+P encaps/decaps functionality. Ideally the CPE also provides NATing functionality. Further, it is expected that at least another component in the ISP network provides the corresponding A+P functionality, and hence is able to establish an A+P subsystem with the CPE. This device is referred to as A+P router or port-range router (PRR), and could be located close to PE routers. The core of the network MUST support the tunneling protocol (which SHOULD be IPv6, as per Constraint 10) but MAY be another tunneling technology when necessary. In addition, we do not wish to restrict any initiative of customers who might want to run an A+P-capable network on or behind their CPE. To satisfy both Constraints 1 and 3 unmodified legacy hosts should keep working seamlessly, while upgraded/new end-systems should be given the opportunity to exploit enhanced features.

4.2. A+P for Mobile Providers

In the case of mobile service provider the situation is slightly different. The A+P border is assumed to be the gateway (e.g., GGSN/PDN GW of 3GPP, or ASN GW of WiMAX). The need to extend the address is not within the provider network, but on the edge between the mobile phone devices and the gateway. While desirable, IPv6 connectivity may or may not be provided.

For mobile providers we use the following terms and assumptions:

1. Provider Network (PN)

2. Gateway (GW)

3. Mobile Phone device (phone)

4. Devices behind phone, e.g., laptop computer connecting via phone to Internet.

We expect that the gateway has a pool of IPv4 addresses and is always in the data-path of the packets. Transport between the gateway and phone devices is assumed to be an end-to-end layer-2 tunnel. We assume that phone as well as gateway can be upgraded to support A+P. However, some applications running on the phone or devices behind the phone (such as laptop computers connecting via the phone), are not expected to be upgraded. Again, while we do not expect that devices behind the phone will be A+P aware/upgraded we also do not want to hinder their evolution. In this sense the mobile phone would be comparable to the CPE in the broadband provider case; the gateway to the PRR/LSN box in the network of the broadband provider.

4.3. A+P from the provider network perspective

ISPs suffering from IPv4 address space exhaustion are interested in achieving a high address space compression ratio. In this respect, an A+P subsystem allows much more flexibility than traditional NATs: the NAT can be placed at the customer, and/or in the provider network. In addition hosts or applications can request ports and thus have untranslated end-to-end connectivity.

```
                 +----------------------------+
       private    | +-------+  A+P-in  +-----+ |   dual-stacked
      (RFC1918) --|-| CPE  |==-IPv6-==| PRR |-|-- network
        space     | +-------+  tunnel  +-----+ |   (public addresses)
                  |    ^              +-----+ |
                  |    |  IPv6-only   | LSN | |
                  |    |   network    +-----+ |
                  +----+----------------- ^ --+
                       |                  |
                  on customer        within provider
             premises and control      network
```

A simple A+P subsystem example

Figure 6

Consider the deployment scenario in Figure 6, where an A+P subsystem is formed by the CPE and a port-range router (PRR) within the ISP core network, preferably close to the customer edge, and represents the border from where on packets are forwarded based on address and port. The provider MAY deploy a LSN co-located with the PRR to handle packets that have not been translated by the CPE. In such a configuration, the ISP allows the customer to freely decide whether the translation is done at the CPE or at the LSN. In order to establish the A+P subsystem, the CPE will be configured automatically (e.g. via a signaling protocol, that conforms to the requirements stated above).

Note that the CPE in the example above is only provisioned with an IPv6 address on the external interface.

```
    +------------ IPv6-only transport ------------+
    | +---------------+ |              |          |
    | |A+P-application| |  +---------+  |  +-----+ |   dual-stacked
    | | on end-host   |=|==| CPE w/ |==|==| PRR |-|-- network
    | +---------------+ |  +---------+  |  +-----+ |   (public addresses)
    +---------------+   |  +---------+  |  +-----+ |
      private IPv4 <-*--+->| NAT    |  |  | LSN | |
      address space   \ |  +---------+  |  +-----+ |
      for legacy       +|--------------|----------+
        hosts           |              |
                        |              |
      end-host with     |  CPE device  |  provider
        upgraded        |  on customer |  network
       application      |   premises   |
```

An extended A+P subsystem with end-host running A+P-aware applications

Figure 7

Figure 7 shows an example of how an upgraded application running on a legacy end-host can connect. The legacy host is provisioned with a private IPv4 address allocated by the CPE. Any packet sent from the legacy host will be NATed either at the CPE (if configured to do so), or at the LSN (if available).

An A+P-aware application running on the end-host MAY use the signaling described in Section 3.1 to connect to the A+P-subsystem. In this case, the application will be delegated some space in the A+P address realm, and will be able to contact the external realm (i.e., the public Internet) without the need for translation.

Note that part of A+P signaling is that the NATs are optional. However, if neither the CPE nor the PRR provides NATing functionality, then it will not be possible to connect legacy end-hosts.

To enable packet forwarding with A+P, the ISP MUST install at its A+P border a PRR which encaps/decaps packets. However, to achieve a higher address space compression ratio and/or to support CPEs without NATing functionality, the ISP MAY decide to provide an LSN as well. If no LSN is installed in some part of the ISP's topology, all CPE in that part of the topology MUST support NAT functionality. For reasons of scalability, it is assumed that the PRR is located within the access-portion of the network. The CPE would be configured automatically (e.g. via an extended DHCP or NAT-PMP, which has the signaling requirements stated above) with the address of the PRR, and

if a LSN is being provided or not. Figure 6 illustrates a possible deployment scenario.

4.4. Dynamic allocation of port ranges

Allocating a fixed number of ports to all CPE may lead to exhaustion of ports for high usage customers. This is a perfect recipe for upsetting more demanding customers. On the other hand, allocating to all customers ports sufficient to match the needs of peak users will not be very efficient. A mechanism for dynamic allocation of port ranges allows the ISP to achieve two goals; a more efficient compression ratio of number of customers on one IPv4 address and, on the other hand, not limiting the more demanding customers' communication.

Additional allocation of ports, or port ranges may be made after an initial static allocation of ports.

The following mechanism applies to NAT functionality in CPE only: If a customer has an arrangement with the ISP for well-known ports, and the PRR allocates to this CPE WKP range, this range may be used for end-to-end communications to a server behind CPE with public IP address or if customer configures so for inbound NAT (1:1 or port forwarding). This function has a fixed range of ports and is not considered in the dynamic pool allocation mechanism. On the other hand, if customer configures the NAT function to access the Internet from a private address pool behind the CPE, this mechanism is automatically applied. NAT keeps track of translation tables, so only a small "daemon" needs to be developed and implemented by the CPE manufacturer to keep track of allocated ranges of ports and how many are used. In the case of 90% usage, the dynamic allocation daemon could signal to the PRR the need for additional ports. A downside of this mechanism is that port allocation to a CPE might get quite large without an additional mechanism that would return unused port ranges back to the PRR's pool. This may be dealt with by requiring the NAT to sequentially allocate ports for translation and reallocate to new requests and released ports. So the use of ports is controlled and unfragmented ranges may be returned to pool. An other, not so pretty, way is to reset the additional allocations to 0 every 24 hours, and leave only the first allocation. Additional allocations would be requested by mechanism in a very short time, leaving the customer unlikely to notice the event.

The mechanism would prefer allocations of port ranges from the same IP address as the initial allocation. If it is not possible to allocate an additional port range from the same IP, then mechanism can allocate a port range from another IP within the same subnet. With every additional port range allocation, the PRR updates its

routing table. The mechanism for allocating additional port ranges may be part of normal signaling that is used to authenticate CPE to ISP.

The ISP controls the dynamic allocation of port ranges by the PRR by setting the initial allocation size and maximum number of allocations per CPE, or the maximum allocations per subscription, depending on subscription level. There is a general observation that the more demanding customer uses around 1024 ports when heavily communicating. So, for example, a first suggestion might be 128 ports initially and then dynamic allocations of ranges of 128 ports up to 511 more allocations maximum. A configured maximum number of allocations could be used to prevent one customer acting in distructive manner should they become infected. The maximum number of allocations might also be more finely grained, with parameters of how many allocations a user may request per some time frame. If this is used, evasive applications may need to be limited in their bad behavior, for example one additional allocation per minute would considerably slow a port request storm.

There is likely no minimum request size. This is because A+P-aware applications running on end-hosts MAY request a single port (or a few ports) for the CPE to be contacted on (e.g., VoIP clients register a public IP and a single delegated port from the CPE, and accept incoming calls on that port). The implementation on the CPE or PRR will dictate how to handle such requests for smaller blocks: For example, half of available blocks might be used for "block-allocations", 1/6 for single port requests, and the rest for NATing.

Another possible mechanism to allocate additional ports is UPnP/NAT-PMP (as defined in Section 3.1), if applications behind CPE support it. In case of the LSN implementation (DS-Lite), as described in the A+P overall architecture section, signaling packets are simply forwarded by the CPE to the LSN and back to the host running the application which requested the ports, and PRR allocates requested port to appropriate CPE. The same behavior may be chosen with AFTR, if requested ports are outside of static initial port allocation. If a full A+P implementation is selected, than UPnPv2/NAT-PMP packets are accepted by the CPE, processed, and the requested port number is communicated through normal signaling mechanism between CPE and PRR tunnel endpoints (DHCP or IPCP).

4.5. Overall A+P architecture

```
                           A+P architecture

         IPv4         Full-A+P          AFTR             CGN
          |              |               |                |
   <-- Full IPv4 ---- Port range ---- Port range  ---- Provider --->
       allocated      & dynamic         & LSN          NAT ONLY
                      allocation      (NAT on CPE      (No mechanism)
       (no NAT)      (NAT on CPE)     and on LSN)      for customer to
                                                       bypass CGN)
```

Figure 8: A+P overall architecture

The A+P architecture defines various options to be deployed within an ISP. Figure 8 shows the spectrum of deployment options. On the far left today's status-quo, an IPv4 address unrestricted with full port-range. Full-A+P, refers to a port-range allocation from the ISP. The customer must operate A+P-aware devices and no NATing functionality is provided by the ISP. AFTR, such as DS-Lite [I-D.ietf-softwire-dual-stack-lite], is a hybrid. There is NAT present in the core (in this draft referred to as LSN), but the user has the option to "bypass" that NAT in one form or an other, for example via A+P, NAT-PMP, etc... Finally, a provider only CGN, will place a NAT in the providers core and does not allow the customer to "bypass" the translation process or modify ALGs on the NAT. The customer is provider-locked. Note as well that all options (besides full IPv4) require some form of tunneling mechanism (e.g., 4in6) and a signaling mechanism (see Section 3.1).

4.6. Example of A+P-forwarded packets

This section provides a detailed example of A+P setup, configuration, and packet flow from an end-host behind an A+P upgraded provider to any host in the IPv4 Internet, and how the return packets flow back. The following example discusses an A+P-unaware end-host, where the NATing is done at the CPE. Figure 9 illustrates how the CPE receives an IPv4 packet from the end-user device. We first describe the case where the CPE has been configured to provide the NAT functionality (e.g., by the customer via interaction via a website, or via automatic signaling). In the following, we call a packet which is translated at the CPE an A+P-forwarded packet, an analogy with the port-forwarding function employed in today's CPEs. Upon receiving a packet from the internal interface, the CPE NATs it and forwards it to the PRR. The NAT on the CPE is assumed to store the 5-tuple (source_IPv4, source_port, destination_IPv4, destination_port,

tunnel-interface).

When the PRR receives the A+P-forwarded packet, it de-capsulates the inner IPv4 packet and it checks the source address and port. If the source address and port match the CPE's A+P address, then the PRR simply forwards the decapsulated packet onward. This is always the case for A+P-forwarded packets. Otherwise, the PRR assumes that the packet is not A+P-forwarded, sl passes it to the LSN function, which in-turn NATs the packet and then releases it into the Internet. Figure 9 shows the packet flow for an outgoing A+P-forwarded packet.

```
                   +-----------+
                   |    Host    |
                   +-----+-----+
                      |  |  10.0.0.2
      IPv4 datagram 1 |  |
                      |  |
                      v  |  10.0.0.1
               +---------|---------+
               |CPE      |         |
               +--------|||--------+
                      | |||     a::2
                      | ||| 12.0.0.3 (100-200)
       IPv6 datagram 2| |||
                      | |||<-IPv4-in-IPv6
                      | |||
                 -----|-|||-------
               /      | |||       \
              |  ISP access network |
               \      | |||       /
                 -----|-|||-------
                      | |||
                      v |||     a::1
               +--------|||--------+
               |PRR     |||        |
               +---------|---------+
                      |  |  12.0.0.1
      IPv4 datagram 3 |  |
                 -----|--|--------
               /      |  |         \
              |   ISP network /     |
               \      Internet     /
                 -----|--|--------
                      |  |
                      v  | 128.0.0.1
                   +-----+-----+
                   | IPv4 Host |
                   +-----------+
```

Figure 9: Forwarding of Outgoing A+P-forwarded Packets

Datagram	Header field	Contents
IPv4 datagram 1	IPv4 Dst	128.0.0.1
	IPv4 Src	10.0.0.2
	TCP Dst	80
	TCP Src	8000
---------------	------------	---------------------------
IPv6 Datagram 2	IPv6 Dst	a::1
	IPv6 Src	a::2
	IPv4 Dst	128.0.0.1
	IPv4 Src	12.0.0.3
	TCP Dst	80
	TCP Src	100
---------------	------------	---------------------------
IPv4 datagram 3	IPv4 Dst	128.0.0.1
	IPv4 Src	12.0.0.3
	TCP Dst	80
	TCP Src	100

Datagram header contents

An incoming packet undergoes the reverse process. When the PRR receives an IPv4 packet on an external interface, it first checks whether the destination port number falls in a delegated range or not. If the address space was delegated, then PRR encapsulates the incoming packet and forwards it through the appropriate tunnel for that IP/port range. If the address space was not-delegated the packet would be handed to the LSN to check if a mapping is available.

Figure 10 shows how an incoming packet is forwarded, under the assumption that the port number matches the port range which was delegated to the CPE.

```
                   +-----------+
                   |    Host   |
                   +-----+-----+
                      ^  |  10.0.0.2
      IPv4 datagram 3 |  |
                      |  |
                      |  |  10.0.0.1
               +---------|---------+
               |CPE      |         |
               +--------|||--------+
                      ^ |||     a::2
                      | ||| 12.0.0.3 (100-200)
       IPv6 datagram 2| |||
                      | |||<-IPv4-in-IPv6
                      | |||
                 -----|-|||-------
               /      | |||        \
              | ISP access network  |
               \      | |||        /
                 -----|-|||-------
                      | |||
                      | |||     a::1
               +--------|||--------+
               |PRR     |||        |
               +---------|---------+
                      ^  |  12.0.0.1
      IPv4 datagram 1 |  |
                 -----|--|--------
               /      |  |         \
              |  ISP network /      |
               \      Internet     /
                 -----|--|--------
                      |  |
                      |  | 128.0.0.1
                   +-----+-----+
                   | IPv4 Host |
                   +-----------+
```

          Figure 10: Forwarding of Incoming A+P-forwarded Packets

Datagram	Header field	Contents
IPv4 datagram 1	IPv4 Dst	12.0.0.3
	IPv4 Src	128.0.0.1
	TCP Dst	100
	TCP Src	80
---------------	------------	---------------------------
IPv6 Datagram 2	IPv6 Dst	a::2
	IPv6 Src	a::1
	IPv4 Dst	12.0.0.3
	IP Src	128.0.0.1
	TCP Dst	100
	TCP Src	80
---------------	------------	---------------------------
IPv4 datagram 3	IPv4 Dst	10.0.0.2
	IPv4 Src	128.0.0.1
	TCP Dst	8000
	TCP Src	80

Datagram header contents

Note that datagram 1 travels untranslated up to the CPE, thus the customer has the same control over the translation as it has today where s/he has an home gateway with customizable port-forwarding.

4.7. Forwarding of standard packets

Packets for which the CPE does not have a corresponding port forwarding rule are tunneled to the PRR which provides the LSN function. We underline that the LSN MUST NOT use the delegated space for NATting. See [I-D.ietf-softwire-dual-stack-lite] for network diagrams which illustrate the packet flow in this case.

4.8. Handling ICMP

ICMP is problematic for all NATs, because it lacks port numbers. A+P routing exacerbates the problem.

Most ICMP messages fall into one of two categories: error reports, or ECHO/ECHO reply (commonly known as "ping"). For error reports, the offending packet header is embedded within the ICMP packet; NAT devices can then rewrite that portion and route the packet to the actual destination host. This functionality will remain the same with A+P; however, the PRR will need to examine the embedded header to extract the port number, while the A+P gateway will do the necessary rewriting.

ECHO and ECHO reply are more problematic. For ECHO, the A+P gateway device must rewrite the "Identifier" and perhaps "Sequence Number" fields in the ICMP request, treating them as if they were port numbers. This way, the PRR can build the correct A+P address for the returning ECHO replies, so they can be correctly routed back to the appropriate host in the same way as TCP/UDP packets. (Pings originated from an external domain/legacy Internet towards an A+P device are not supported.)

4.9. Limitations of the A+P approach

One limitation that A+P shares with any other IP address-sharing mechanism is the availability of well-known ports. In fact, services run by customers that share the same IP address will be distinguished by the port number. As a consequence, it will be impossible for two customers who share the same IP address to run services on the same port (e.g., port 80). Unfortunately, working around this limitation usually implies application-specific hacks (e.g., HTTP and HTTPS redirection), discussion of which is out of the scope of this document. Of course, a provider might charge more for giving a customer the well-known port range, 0..1024, thus allowing the customer to provide externally available services. Many applications require the availability of well known ports. However, those applications are not expected to work in A+P environment unless they can adapt to work with different ports. However, such application do not work behind today's NATs either.

Another problem which is common to all NATs is coexistence with IPsec. In fact, a NAT which also translates port numbers prevents AH and ESP from functioning properly, both in tunnel and in transport mode. In this respect, we stress that, since an A+P subsystem exhibits the same external behavior as a NAT, well-known workarounds (such as [RFC3715]) can be employed.

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

The primary security issue any time a NAT is mentioned is the implicit firewall provided by a NAT. Any proposal to eliminate NATs raises the spectre of insecure hosts lying naked before a hostile

Internet. For a number of reasons, we do not think this is a serious issue here. If nothing else, NATs are not really security devices; their protective value is limited.

A NAT owned by a customer, whether a home consumer or a large enterprise, is under the control of that customer. All machines on the customer's side of the NAT have unfettered access to other machines on the same side; generally, this is what is desired. A+P NATs do not change this, as the customer has still controls what is being NATed. LSN does not change the access property, either. However, with a CGN without A+P there are *many* machines on the inside of the translation, not all of which are in the customer's administrative domain. Unless other firewall mechanisms are employed, LSNs create added risk of unauthorized access.

By contrast, the protection scope of an A+P NAT is, by definition, at the boundary to the customer network. The access properties are thus precisely what traditional NATs have provided.

There is one notable exception to this point. Inbound packets addressed to the assigned port number range are passed through unchanged, even if no outbound packets were sent to the originator. While this allows customers to run their own servers on certain ports, it also allows attackers to probe these servers without the protection provided today by provider-supplied NAT boxes. The issue is not that internal machines are addressable -- that is an inevitable corollary to servers being run -- but that it may represent a change from today's behavior. Furthermore, the effect on the customer varies greatly, depending on what port number range they are assigned; someone who is assigned 0-4K derives more benefit and runs more risk than someone who is assigned 48K-52K, since the latter is in the IANA-assigned dynamic port range.

A useful middle ground would be provision of a customer-controllable switch in the CPE to control what happens to such packets. If filtering is to be done, state must be kept, which might be costly. This suggests that perhaps it should only be done in the CPE if it is replacing current CPE that provides NAT functionality. If applications on end-hosts installed A+P gateways, they might open up ports untranslated.

Note that, regardless of the existence of such an option, the A+P gateway will need customer-controllable port number-mapping capability, as most customers will not be assigned a range which corresponds to the servers they wish to run.

With CGN/LSNs, tracing hackers, spammers and other criminals will be extremely difficult, requiring logging, recording, and storing of all

connection based mapping information. The need for storage implies a tradeoff. On one hand, the LSNs can manage addresses and ports as dynamically as possible, in order to maximize aggregation. On the other hand, the more quickly the mapping between private and public space changes, the more information needs to be recorded. This would not only cause concern for law enforcement services, but also for privacy advocates.

A+P offers a better set of tradeoffs. All that needs to be logged is the allocation of a range of port numbers to a customer. By design, this will be done rarely, improving scalability. If the NAT functionality is moved further up the tree, the logging requirement will be as well, increasing the load on one node, but giving it more resources to allocate to a busy customer, perhaps decreasing the frequency of allocation requests.

The other extreme is A+P NAT on the customer premises. Such a node would be no different than today's NAT boxes, which do no such logging. We thus conclude that A+P is no worse than today's situation, while being considerably better than CGNs.

7. Authors

This document has 8 primary authors, which is not allowed in the header of Internet-Drafts. This is the list of actual authors of this document.

Gabor Bajko
Nokia
Email: gabor(dot)bajko(at)nokia(dot)com

Steven M. Bellovin
Columbia University
1214 Amsterdam Avenue
MC 0401
New York, NY 10027
US
Phone: +1 212 939 7149
Email: bellovin@acm.org

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US
Phone: +1 206 780 0431 x1
Email: randy@psg.com

Luca Cittadini
Universita' Roma Tre
via della Vasca Navale, 79
Rome, 00146
Italy
Phone: +39 06 5733 3215
Email: luca.cittadini@gmail.com

Alain Durand
Comcast
1 Comcast Center
Philadelphia, PA
US
alain_durand@cable.comcast.com

Olaf Maennel
Loughborough University
Department of Computer Science - N.2.03
Loughborough
United Kindom
Phone: +44 115 714 0042
Email: o@maennel.net

Teemu Savolainen
Nokia
Hermiankatu 12 D
TAMPERE, FI-33720
Finland
Email: teemu.savolainen@nokia.com

Jan Zorz
go6.si
Frankovo naselje 165
Skofja Loka 4220
Slovenia
Phone: +38659042000
Email: jan@go6.si

8. Acknowledgments

The authors wish to especially thank Remi Despres, and Pierre Levis for their help on the development of the A+P approach. We also thank David Ward for review, constructive criticism, and interminable questions, and Dave Thaler for useful criticism on "stackable" A+P gateways. We would also like to thank the following persons for their feedback on earlier versions of this work: Rob Austein, Gert

Doering, Dino Farinacci, Russ Housley, and Ruediger Volk.

9. References

9.1. Normative References

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

9.2. Informative References

[BCP38] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, May 2000.

[I-D.bajko-pripaddrassign] Bajko, G., Savolainen, T., Boucadair, M., and P. Levis, "Port Restricted IP Address Assignment", draft-bajko-pripaddrassign-01 (work in progress), March 2009.

[I-D.boucadair-dhcpv6-shared-address-option] Boucadair, M., Levis, P., Grimault, J., Savolainen, T., and G. Bajko, "Dynamic Host Configuration Protocol (DHCPv6) Options for Shared IP Addresses Solutions", draft-boucadair-dhcpv6-shared-address-option-00 (work in progress), May 2009.

[I-D.boucadair-port-range] Boucadair, M., Levis, P., Bajko, G., and T. Savolainen, "IPv4 Connectivity Access in the Context of IPv4 Address Exhaustion: Port Range based IP Architecture", draft-boucadair-port-range-02 (work in progress), July 2009.

[I-D.boucadair-pppext-portrange-option] Boucadair, M., Levis, P., Grimault, J., and A. Villefranque, "Port Range Configuration Options for PPP IPCP", draft-boucadair-pppext-portrange-option-01 (work in progress), July 2009.

[I-D.ietf-softwire-dual-stack-lite] Durand, A., Droms, R., Haberman, B., Woodyatt, J., Lee, Y., and R. Bush, "Dual-stack lite broadband deployments post IPv4 exhaustion", draft-ietf-softwire-dual-stack-lite-01 (work in progress), July 2009.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC3715] Aboba, B. and W. Dixon, "IPsec-Network Address Translation (NAT) Compatibility Requirements", RFC 3715, March 2004.

Author's Address

Randy Bush (editor)
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Phone: +1 206 780 0431 x1
Email: randy@psg.com