

# A Better Approach than Carrier-Grade-NAT

Olaf Maennel   Randy Bush   Luca Cittadini   Steven M. Bellovin

T-Labs  
Berlin, Germany  
olaf@maennel.net

IJ  
Tokyo, Japan  
randy@psg.com

Universita' Roma Tre  
Rome, Italy  
luca.cittadini@gmail.com

Columbia University  
New York, USA  
smb@cs.columbia.edu

Technical Report CUCS-041-08

## ABSTRACT

We are facing the exhaustion of newly assignable IPv4 addresses. 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, for which many solutions have been proposed. Some prominent ones target towards carrier-grade-NATs (CGN), e.g. [1], which we feel is a bad idea. Instead, we propose using specialized NATs at the edge that treat some of the port number bits as part of the address.

## 1. INTRODUCTION

It appears inevitable that the Internet will become a polymorphic architecture, many different technologies will have to co-exist in the future. This becomes obvious when looking at IPv4 exhaustion, while there is only marginal IPv6 deployment. Operators are getting worried about extremely large routing tables, an Internet core that does not really want to know about all those bright traffic engineering ideas from the edge, and the increased mobility of end-devices. This names but a few of the issues the IRTF Routing Research Group (RRG)<sup>1</sup> is trying to tackle.

The looming exhaustion of the free IANA IPv4 pool creates even more urgent problems. Many large Internet Service Providers (ISPs) face the problem, that their networks' customer edges are so large that, even giving the "front" of each customer premises equipment (CPE) only one single IPv4 address, they need two to five /8s of IPv4 space [2]. However, it is highly unlikely that they would be allocated that much public IPv4 address space. Therefore ISPs have to come-up with something more ingenious. Deploying a NAT is a direct consequence of the design of a new protocol which is incompatible on the wire, there is not the slightest compatibility mode. We may not like this, and we certainly do not, but NATs are inevitable.

The approach ISPs are testing is being called Carrier Grade NAT (CGN). It is essentially a number of IPv4 NATs in the core of their networks and various tunneling and transla-

tion techniques [1]. If the CPE has dual stack, traffic where source and destination is IPv6 would not have to be NATted, but IPv4 would be heavily NATted. We can contrast this to, for example, NAT-PT [3, 4] on the CPE, which would probably scale to the needs of even a large non-consumer backbone. But, as we noted above, very large broadband consumer providers would need far too much IPv4 space for the NAT-PT front ends for their large consumer networks.

Our main concern is that the imminent IPv4 address exhaustion is leading operators to deploy extremely damaging technology.

### 1.1 Why Carrier-Grade-NATs are harmful

We have taken up a desperate search for alternatives. The reasons are simple:

"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 and smart edges.

With the smarts at the edges, e.g. NAT-PT, one can easily field new protocols between consenting end-points by just tweaking the NATs at the consenting CPEs, even adding application layer gateways (ALGs) if needed. However, CGN's do not build an Internet walled garden at the edges, they build it by restricting the core.

With NAT in the core, if a customer wants a new application protocol which requires cooperation from the NAT, they get to beg help from the broadband providers' engineers and lawyers, and all other users of carrier grade NATs. This is the ultimate horror the NAT-haters fear, and they are not all that wrong.

One broadband provider has recently received a lot of bad press for just this, though we know that the engineers are very far from those responsible. This shows that all new application protocols have to go through the carrier loving lawyers to be allowed to be handled by the NATs in their core. Today's NATs are typically mitigated by ALG's of which the customer has some degree of control, e.g. port forwarding or UPnP. However, this is **not** expected to work anymore with CGN's. CGN proposals admit that it is not expected that applications that require specific port assignment or port mapping from the NAT box will keep working [1].

<sup>1</sup><http://www.irtf.org/charter?gtype=rg&group=rrg>

We believe this is not an option and that the end-user must have the ability to control their own ALGs. So, if someone wants to deploy a new application, they can talk to the broadband providers' lawyers or do it over HTTP, pick your poison.

And remember that, as IPv6 deploys, and we want to have one Internet, i.e. IPv4 nodes talking freely with IPv6 nodes, then translation must be done somewhere. The challenge is whether someone can figure out a scheme where it is done for these large networks? We believe it should be at the customer edge, not in the core.

Another issue with CGN's is scalability. ISPs face a tension in the placement within their network: to achieve the desired effect means to aggregate as much as possible, but on the other hand this creates a massive state problem. To reduce the state, the placement location could be somewhere closer to the edge, where the benefits are limited.

It is not clear how a CGN should maintain per-session state in a scalable manner. This is particularly relevant given that each customer is very likely to open many TCP connections in parallel. 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, and this makes optimally placing a CGN a hard engineering problem.

Tracing back hackers, spammers and other criminals will be impossible, unless all the connection based mapping information is recorded and stored. This would cause not only concern for law enforcement services, but also for privacy advocates. Which brings us to other security related problems with CGN's in the next section.

## 1.2 Security of CGN

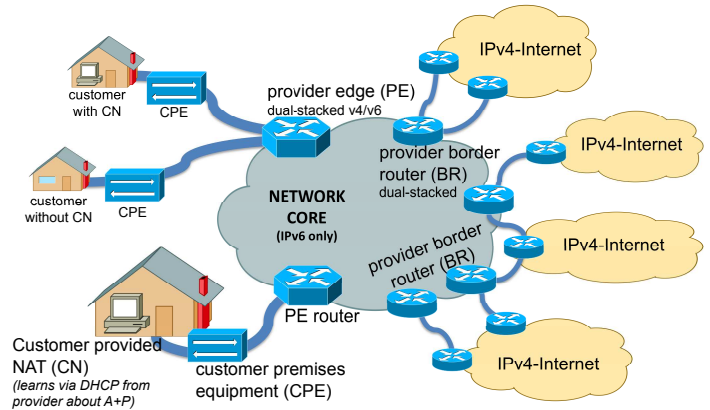
NATs frequently need to initiate translation for secondary port numbers. This may be a decision based on packet inspection (i.e., looking for PORT commands in FTP [5] sessions), or it may rely on explicit signaling from the end host via protocols such as UPnP. Either way, CGNs pose a security threat and/or an administrative nightmare.

The issue is proper authentication of such requests. Most UPnP devices do not implement any security features. Even if they did, there would be no way to administer the security mechanism. Every end-user device would have to have a secret corresponding to some authentication field in the CGN. End users will not set these up properly; providers do not want to maintain such a database.

Decisions made based on packet inspection are just as problematic. A request from one customer could easily request opening a port for another customer's addresses, similar to the Java-based attack described by Martin et al [6].

## 2. PROPOSED SOLUTION

As the specific problem is insufficient IPv4 address space to number the IPv4-speaking customers, we propose to extend the IPv4 address space by assigning to each customer



**Figure 1: Proposed architecture: CN and CPE are on customer premises and learn port-range allocation via DHCP; PE is dual-stack IPv4/IPv6 and connect customer via a direct interface; BR's are also dual-stack and terminate the IPv4 tunnels.**

a single IPv4 address which is extended by “stealing” bits from the port number in the TCP/UDP header, leaving the applications a reduced fixed range of ports. 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, and broadband consumers are not anticipated to deploy a massive number of applications over IPv4 (if they did, CGN would be even more damaging than this bit-stealing proposal). Assuming we could limit the applications' port addressing to 6 (or 7) bits, we can increase the effective size of an IPv4 address by 10 (or 9) additional bits. In this scenario, 1024 (or 512) customers could be multiplexed on the same IPv4 address, while allowing them a fixed range of 64 (or 128) ports. We call this *extended addressing* or *A+P* (address plus port) addressing.

### 2.1 Changes required to the network

Figure 1 summarizes the devices involved in this solution:

- Customer Premises Equipment (CPE), the cable/DSL modem
- Customer-Provided-NAT (CN), *optional*
- Provider Edge Router (PE), AKA customer aggregation router
- Provider Border Router (BR), provider's edge to other providers
- Network Core Routers (Core), provider routers not PE or BR

*Customer-Premises-Equipment.* As the customer's hosts should be unaware of the restricted range of ports and the extended A+P addressing scheme, translation would be done at the border between the customer and the provider. In the most common case, this is the provider provisioned cable or DSL modem on the customer's premises into which the customer plugs their single computer or a LAN. This CPE would be aware of the A+P extended addressing, and would

provide the A+P NAT function between the customer's LAN and the provider.

This would require modification of current CPE, but the CGN draft [1] says "It is expected that the home gateway is either software upgradable, replaceable or provided by the service provider as part of a new contract."

The customer premises equipment would be configured, hopefully automatically, with

- IPv4 and/or IPv6 addressing for the customer's LAN
- The IPv4 A+P extended address for the WAN side to connect to the provider,
- An IPv6 address for the WAN side to connect to the provider, and
- The range of port number to use on the WAN side.

*Customer-Provided-NAT.* Alternatively, as occasionally happens today, the customer could provide the A+P NAT and the CPE would then be configured as a simple cable/DSL modem. This customer A+P NAT would be configured with the IPv4 address and port-range allocated to the customer. This could be done, for example, via a vendor or other extension to DHCP.

The customer NAT is entirely *optional*. The customer does not have to operate such a device. If they do not, then the provider installed CPE could handle the mappings. A mixture of CPE and CN device is also possible, where the customer gets full control over the CPE via an administrative login.

*Provider-Edge Router.* During transition, customers with legacy CPE and no CN would have the A+P NAT function provided by the provider's customer aggregation, AKA PE, router. Customer packets would be A+P NATted as soon as they reached the PE. If we assume only layer 2 devices which connect directly to an interface of the PE, then there should be no problems for the customer to be unaware of the restricted port range.

In a sense this is a carrier-grade-NAT. However, two important differences apply: (a) the customer has the possibility to operate on non-NATed-ports and is aware of which ports will be NATed and which are not; and (b) the carrier-grade-NAT is very close to the customer (e.g., on the PE router), so should scale well as a NAT.

But note that having the A+P NAT on the PE router is effectively the same walled garden effect that a CGN would cause. Therefore this is only a transition mechanism and a method for connecting customers who do not need or request control of their own application destinies.

*Provider Border Routers.* Routers at the provider's edge which faces other providers need to be aware of the extended A+P IPv4 addresses. They must have the ability to forward packets to the PE based on IPv4 address and port.

For example, the provider network could use IPv6 as a

tunneling mechanism. The CPE or PE routers would encapsulate the A+P pseudo address within an IPv6 address using a well-known IPv6 prefix. Then the core would route on the IPv6 address. The border routers would recognize the well-known IPv6 prefix and decapsulate and forward the IPv4 address.

Thus the provider's network could be IPv6 only, or any other layer 3/2.5 protocol.

*Network Core Routers.* If the tunneling technique within the provider is chosen appropriately, it would not be required that the network's core routers are capable of understanding A+P extended IPv4 addressing. In fact, as doing so would require that the core deploy IPv4 all the way to the PE routers, and the original problem was insufficient IPv4 space, we assume that IPv6 or other non-IPv4 tunneling will be preferred.

Note that, in the IPv6 tunneling technique we use as an example, all customers hosted on the same IPv4 address do not need to be attached to the same PE router. Normal IPv6 routing protocols would be used, and moving one customer from a specific A+P pool could be done by the PE announcing a longer IPv6 prefix.

## 2.2 Design of the A+P NAT Device

There are a number of delicate design choices for the A+P NAT device. We present our preferred solution here. Other options are discussed in Appendix 5.

Legacy hosts would send IPv4 packets from any port(s). We are not expecting to change end-hosts; therefore we require some kind of NAT. However, one of the basic requirements is that the customer must be able to run their own servers and NATs. This leads to several constraints:

1. We want to enforce the analog of BCP 38 [7], for many obvious reasons. This means that no packets outside of the assigned address and port number range should leave the PE for the network.
2. We want minimal configuration. There should be no need for the customer to tell the ISP that they have purchased an A+P-grade home NAT.
3. We must support unmodified computers and NATs.
4. We want the PE router to be as accommodating as possible to strange protocols it knows nothing about. It may do its own packet snooping and/or ALGs for things it knows about (i.e., FTP, SIP, Skype), but should leave it to the customer's box to handle World of Warcraft, Second Afterlife, or what have you.
5. Conversely, if the customer's box has done something, it should be left alone; it should not be retranslated.

These principles lead us to the following design:

1. The PE should discard any outbound packets that don't originate from the proper IP address. (Constraint 1)

2. An A+P gateway should include some option in the DHCP request message, to inform the PE router of its abilities. (Constraint 2)
3. If no A+P signaling was done, the PE router should perform NATting, including whatever ALG functions it can. (Constraints 3 and 4)
4. The PE router should leave intact any packets from the proper address and port range. (Constraints 4 and 5)

Note that a customer with no NAT or with a non-A+P NAT may emit packets within the proper port range by accident, thus accidentally violating part of point 4 above. We solve that by DHCP-based signaling from the A+P box: the A+P option in the DHCP request tells the PE that a customer-provided box will do all NATting according to this design. In that case, the primary function of the PE router is to enforce restrictions on port numbers in outbound packets.

We leave unspecified for now the question of how large a port number range is allocated to each customer. We anticipate that the allocation available to a customer will be determined by ISP-specific policy, perhaps as a function of the fee charged to the customer. If variable allocations are to be supported, i.e., the ability for a customer to request more port numbers (and hence more possible simultaneous connections) at one time and fewer at another, the natural way to signal this is in the DHCP A+P request option.

A more interesting question is how to increase the range dynamically. A simple DHCP release/request cycle could be used, but if the proper adjacent block of port numbers was not available, this would entail tearing down existing connection or reNATting them. The disadvantages of the former are obvious; adopting the latter approach would bring back all of the disadvantages this scheme is intended to avoid. One possible answer is to allocate additional ⟨address, port-range⟩ pairs. We leave this issue for future work.

*IPv6 and mixed V4-V6 traffic.* Note that if IPv4/IPv6 dual stack is provided on the customer’s LAN, IPv6 to IPv6 destinations would be transported untranslated from the customer’s host to the provider’s border with other providers.

If the customer has an IPv6-only LAN, then the device providing A+P translation should also provide NAT-PT service so that the customer could communicate with the IPv4 Internet.

*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; NATs 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 provider’s BR will need to examine the embedded header

to learn with A+P NAT is handling it, while that box will do the necessary rewriting.

ECHO and ECHO reply are more problematic. For ECHO, the border router must rewrite the “Identifier” and perhaps “Sequence Number” fields in the ICMP request, so that returning ECHO REPLY packets may be routed correctly. We suggest to rewrite the information in the sequence number to allow the BR returning ECHO replies to come back to the appropriate host.

*Handling IP fragments.* Much like ICMP packets, IP fragmented packets are renowned to be hard to handle in any address translation mechanism [8]. In fact, only the first IP fragment contains the TCP (UDP) header. This issue is commonly dealt with by keeping additional state at the NAT device which allows fragments to be mapped to the correct TCP (UDP) session. In the A+P NAT solution, fragments coming from the internal domain can be avoided if the core network runs IPv6 only and the PE ensures that no layer-3 fragmentation is performed by the customer equipment. Fragments coming from the external domain are harder to handle. Commercial NATs extract the port number out of the first fragment and keep that information to map subsequent fragments. Moreover, when the first fragment is not the first one to be received at the NAT, the fragment needs to be stored until the port number is known [9].

### 3. RELATED WORK

There is a long history of treating port numbers as part of the network address. It was considered as part of the design of TCP/IP [10]. In the same time frame, Pup [11] and the Xerox Network System architecture [12] included the “socket number” as part of an address; the other two parts were a network number and a -bit host number. However, only the network number was used for routing. Later, Bellovin and colleagues made suggestions that embedded the service in the IP address; see [13] and [14].

The work most closely related to A+P routing is [15]. In it, Hang Zhao, Chi-Kin Chau, and Steven M. Bellovin suggest routing on the ⟨address,port⟩/48 string, i.e., a route per service. Thus, if there is no route advertisement for, say, ⟨A,25⟩/48, every router along the path will decline to forward SMTP packets to host A. However, that work had no notion of address or port number translation.

### 4. ACKNOWLEDGMENTS

This work was partially supported by a gift from Cisco. We also like to thank the following persons for their valuable feedback: Alain Durand, Russ Housley, Hamed Haddadi, and Wolfgang Mühlbauer.

### 5. OPEN QUESTIONS

- How many ports does the end-host need? We should expect a few hundreds of outgoing connections. Incom-

ing connections are instead much less popular (only p2p and something else), so maybe 64 or 128 is a fair deal? Maybe 32 is enough?

- Well known ports are an issue. Customers will not be able to offer services on well known ports. They may or may not get a port-range that overlaps with the well known port-range, but so far this proposal does not address this issue.

## 6. REFERENCES

- [1] A. Durand, "Dual-stack lite broadband deployments post IPv4 exhaustion," Internet draft, draft-durand-dual-stack-lite-00, work-in-progress, July 2008.
- [2] Alain Durand, "Managing 100+ Million IP Addresses," <http://nanog.org/mtg-0606/durand.html>, 2006, NANOG 37.
- [3] G. Tsirtsis and P. Srisuresh, "Network address translation - protocol translation (nat-pt)," RFC 2766, Internet Engineering Task Force, Feb. 2000.
- [4] C. Aoun and E. Davies, "Reasons to move the network address translator - protocol translator (nat-pt) to historic status," RFC 4966, Internet Engineering Task Force, July 2007.
- [5] J. Postel and J. Reynolds, "File transfer protocol," RFC 959, Internet Engineering Task Force, Oct. 1985.
- [6] David Martin, S. Rajagopalan, and Aviel D. Rubin, "Blocking Java applets at the firewall," *Proceedings of the Internet Society Symposium on Network and Distributed System Security*, pp. 16–26, 1997.
- [7] P. Ferguson and D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing," BCP 38, Internet Engineering Task Force, May 2000.
- [8] P. Srisuresh and K. Egevang, "Traditional ip network address translator," RFC 3022, Internet Engineering Task Force, Jan. 2001.
- [9] J. Doyle, *Routing TCP/IP Volume I (CCIE Professional Development)*, Cisco Press, 1998.
- [10] Vint Cerf, , 2004, Private conversation.
- [11] David R. Boggs, John F. Shoch, Edward A. Taft, and Robert M. Metcalfe, "Pup: An internetwork architecture," *IEEE Transactions on Communications*, vol. COM-28, no. 4, pp. 612–624, April 1980.
- [12] Xerox System Integration Standard, "Internet transport protocols," XSI 028112, Xerox Corporation, December 1981.
- [13] S. Bellovin, "On many addresses per host," RFC 1681, Internet Engineering Task Force, Aug. 1994.
- [14] Peter M. Gleitz and Steven M. Bellovin, "Transient addressing for related processes: Improved firewalling by using IPv6 and multiple addresses per host," in *Proceedings of the Eleventh Usenix Security Symposium*, August 2001.
- [15] Hang Zhao, Chi-Kin Chau, and Steven M. Bellovin, "Roff: Routing as the firewall layer," in *New Security Paradigms Workshop*, September 2008, A version is available as Technical Report CUCS-026-08.

## APPENDIX

### A. OTHER DESIGN OPTIONS

There are other design possibilities than the ones we presented in Section 2.2, in particular when handling outbound packets. We list those here.

1. **Ignore/do nothing.** Since a portion of the port number is meant to extend the IPv4 address, allowing customers to produce packets out of their assigned port range should be regarded as letting them "spoof" the source IP address. In particular, return packets will be routed back to a different host! While this might be the most cost-effective way, it certainly has security issues. Even worse, it prevents unaware applications on the customer's side from running properly.
2. **Filter.** If the PE drops all packets that are not originated from the allocated source port range, then some of the security concerns could be reduced. However, the customer would still be unhappy with malfunctioning applications.
3. **Symmetric translation.** The PE acts as a symmetric NAT for every outgoing connection. More precisely, if a customer sends a packet, the PE will pick a port in the customer's assigned port range and create a mapping for the following tuple:  $\langle \text{protocol, mapped source port, destination IP, destination port} \rangle \Rightarrow \langle \text{source IP, source port} \rangle$ . If an inbound packet matches an entry in the mapping table, it will be translated back, and sent to the original port. If there is no entry in the mapping table it will be forwarded directly to the customer without modifications. This behavior can be obtained by slightly modifying standard existing technology (e.g., symmetric NAT for outgoing connections and static port mapping for incoming connections).  
A limitation of this approach arises if the customer is autonomously running a NAT which is unaware of the restricted port range. Those devices are widely deployed in today's Internet and they will create 2 levels of address-translations. However, we expect the customer that is running their own NAT gateway to be able to correctly configure it according to the port range that the customer has been assigned from the provider, thus mitigating problems with the traversal of multiple NATs.
4. **PE ensures source ports are from within the allocated address space, and if not uses NAT to translate towards special "excess" IP addresses.** This solution would allocate pseudo-IPv4 to customers who want to run their own home-gateway-NATs or servers. In addition to this address space the PE would have some extra address space to do the NATs for the customers. In this case the port-

overlap problem discussed above would be fixed. However, additional address space is needed and a majority of customers might not use the specific allocated address space at all.