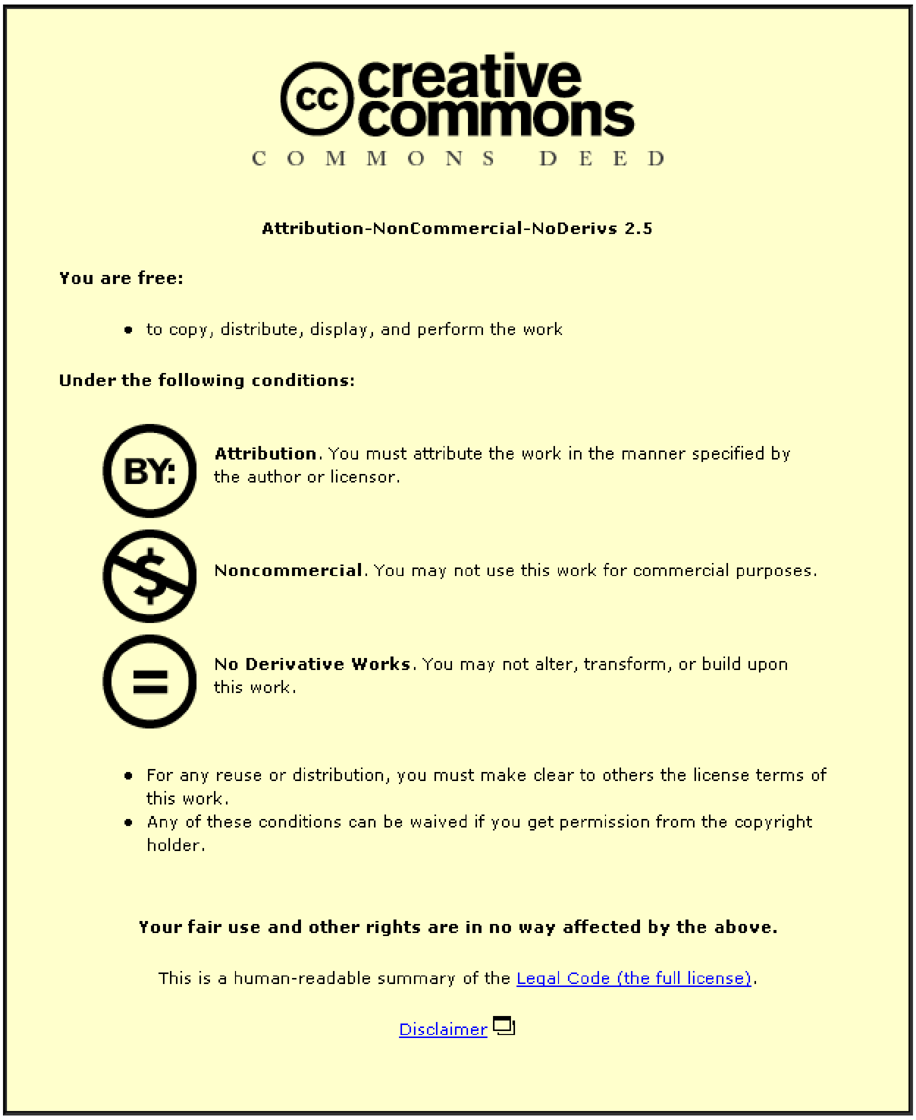

IPv4 Address Sharing Mechanism Classification and Trade-Off Analysis

Nejc Škoberne, Olaf Maennel, Iain Phillips, Randy Bush, Jan Žorž, Mojca Ciglarič

***Abstract*—The growth of the Internet has made IPv4 addresses a scarce resource. Due to slow IPv6 deployment, IANA-level IPv4 address exhaustion was reached before the world could transition to an IPv6-only Internet. The continuing need for IPv4 reachability will only be supported by IPv4 address sharing.**

This paper reviews ISP-level address sharing mechanisms, which allow Internet service providers to connect multiple customers who share a single IPv4 address. Some mechanisms come with severe and unpredicted consequences and all of them come with trade-offs.

We propose a novel classification, which we apply to existing mechanisms such as NAT444 and DS-Lite and proposals such as 4rd, MAP etc.

Our trade-off analysis reveals insights into many problems including: abuse attribution, performance degradation, address and port usage efficiency, direct inter-customer communication and availability.

***Index Terms*—IPv4 address sharing, IPv6 transition, address plus port (A+P), network address translation (NAT), carrier grade NAT (CGN), address family translation.**

I. Introduction

On February 3rd 2011, the Internet Assigned Numbers Authority (IANA) announced that the pool of public IPv4 Internet addresses had become depleted. Consequently Regional Internet Registries (RIRs) were left with only the addresses they had been assigned prior to this date. On April 15th 2011, the Asia-Pacific Network Information Center (APNIC) activated its "last /8 address policy" [1]. Similarly, on September 14th 2012 Réseaux IP Européens Network Coordination Centre (RIPE NCC) activated their last /8 policy. This means that any organization applying to these RIRs for IPv4 address space will receive a maximum allocation of one and only one /22 prefix (1024 IPv4 addresses). Such allocations are too small to satisfy current growth rates.

The only long-term solution to the IPv4 address exhaustion problem is transition to the IPv6 protocol, which enables addressing large numbers of Internet devices [2]. However, today we observe little IPv6 deployment. IPv6 penetration at content providers (top 500 web sites) is about 24% globally [3] as of 03/15/2013, while at the user side is something more than 1% [4] as of 03/15/2013. As IPv4 and IPv6 are incompatible, IPv6 designers envisioned a dual-stack deployment [5], with the aim that by the time the IPv4 address space became depleted, IPv6 would be universally deployed. Unfortunately, this did not happen, though some Internet Service Provider (ISP) backbones have moved to dual-stack in the last few years. Thus, the IPv4 protocol remains the predominant protocol and will do for some time during which the coexistence of both Internets needs to be maintained [6]. Eventually IPv6 will become ubiquitous and IPv4 no longer of interest.

Some ISPs do not have enough IPv4 addresses to provide a dedicated IPv4 address to each customer. To support continued growth, individual IPv4 addresses will have to be shared between multiple customers, which we refer to as "ISP-level address sharing". However, the consequences of deployment of these mechanisms for the Internet users is not well understood.

Unfortunately, ISPs often do not have enough information about the potential consequences of their decisions. E.g.:

- Would the deployment of a double-NAT mechanism prevent Xbox LIVE customers who share the same IP address from playing games online and ultimately lead to loss of many customers and potentially to a bad reputation for that ISP?
- Will cyber-criminals be untraceable, because content providers (today) only log time and IP address of the attacker, but not source ports?
- Does the deployment of a particular mechanism create provider lock-in, so that the customers have to use, say, the Internet TV service of their ISP as competitors' services fail to work?
- Will the End-to-End Principle, which is one of the core principles of the Internet [7], become even more endangered with ISP-level address sharing?
- Will all new protocols have to tunnel over HTTP as this may be the only remaining application-layer protocol which traverses address sharing devices?

This paper presents a systematic approach to classifying and analyzing existing IPv4 address sharing mechanisms. To compare and to understand them, we abstract some of their details and explore the whole solution space. First, we define the classification dimensions and properties. Then, we infer nine classes categorizing existing mechanisms. Similar mechanisms are classified into the same class.

Our main research objective is to propose a classification for IPv4 address sharing mechanisms. We feel the need for such classification is significant: revealing gaps and conflicts, while new Internet Drafts of address sharing proposals keep coming, many of them expiring after a year or two. Other networking research papers focus on these drafts arbitrarily, while they could focus on whole classes instead and thus gain

N. Škoberne is with Viris, Slovenia, e-mail: nejc@viris.si.

O. Maennel, I. Phillips are with Computer Science, Loughborough University, UK, e-mail: olaf@maennel.net, i.w.phillips@lboro.ac.uk

R. Bush is with Internet initiative Japan, e-mail: randy@psg.com

J. Žorž is with Go6 Institute, Slovenia, e-mail: jan@go6.si

M. Ciglarič is with University of Ljubljana, Slovenia, e-mail: mojca.ciglaric@fri.uni-lj.si.

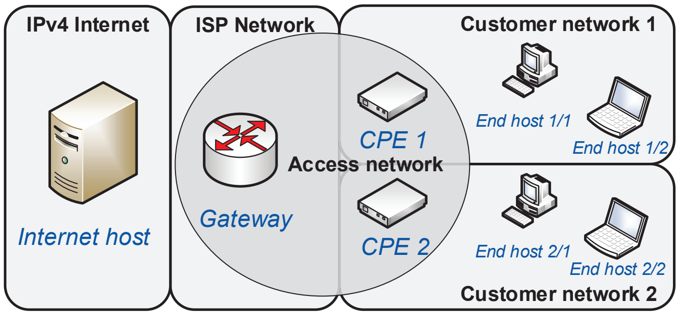

Fig. 1. An ISP topology. For the sake of simplicity, two customer networks are shown, though realistically there could be thousands or millions of them.

more universal value. Additional classes might be defined in the future. Further, our results will inform the design of new address sharing mechanisms.

We consider networks where address sharing must be used, including broadband ISPs providing internet access to large numbers of customers. However, this excludes mobile ISPs even though their number of subscribers have long surpassed the number of wired subscriptions. We believe ISP-level address sharing will have stronger impact on wired users than on mobile ones, as non-mobile end hosts usually have higher requirements, e.g., peer-to-peer networking, than mobile end hosts. Moreover, it is common practice for mobile users use WiFi networks when available, so their device becomes another end host in our topology. We only consider unicast, multicast is out of scope. We use the terms *port* and *flow* in the context of transport-layer protocols like TCP and UDP.

First we review terms we use for the topology of an ISP, see figure 1:

- **ISP network**. The network of the Internet service provider, which also contains the access network.
- **Gateway**. A device in the core of the ISP network which processes customer traffic to and from the Internet.
- **Access network**. The network connecting CPEs and gateways in the ISP network.
- **CPE** (Customer Premises Equipment). A device at customer's premises which processes the traffic between the customer's network and the access network.
- **Customer network**. The network behind the customer's CPE, for which the ISP provides Internet access.
- **End host**. A device desiring access to the IPv4 (and possibly IPv6) Internet residing in the customer's network.

We make some assumptions about the networking topology. First, we do not consider address sharing mechanisms where end host modification is required; this is a realistic requirement as it is infeasible to change deployed hosts, e.g., it was years after the IPv6 RFC was published until Windows XP had production-ready IPv6 support. Second, every customer is assumed to have a CPE, even though some mechanisms allow for connecting end host directly to the access network. CPE, however, may be modified or replaced for the purpose of deploying some mechanism.

This paper has four main contributions. First, it provides a classification of IPv4 address sharing mechanisms by proposing and explaining five dimensions. Each has multiple properties. (Section III-B).

Next, we classify the mechanisms into nine classes and review them. We aim to find general similarities that facilitate understanding and discussion of such technologies (Section III-C).

Third, we discuss the properties of the mechanisms based on the proposed classification. We also identify and describe the most important practical technical issues related to specific properties of each approach (Section IV).

Finally, we present an analysis qualitatively describing the trade-offs between the classification dimensions. The analysis aims at guiding ISPs through their decisions and at providing a grounding for future research in this area (Section V).

II. Background and Related Work

CPEs can share a public IPv4 address in two ways:

1) *Carrier-Grade NAT* (CGN): using a translator with a Network Address and Port Translation (NAPT) function located in the core of the ISP's network, which multiplexes multiple CPEs on a single IPv4 address.
2) *Address-Plus-Port* (A+P): by communicating in a port-restricted manner, where bits from the port field are used to extend the IPv4 address field, i.e., choosing a source port for outgoing packets from a subset of the whole 16-bit port range and receiving incoming packets destined to a port from the same subset. For this to work, the CPEs have to port-restrict outgoing packets and the gateways have to route incoming packets using the destination port.

We only consider A+P CPE and not A+P end hosts. A+P CPE therefore must perform a (port restricted) NAPT function to support multiple end hosts.

In the time of dial-up Internet access, IP addresses were shared over time using Dynamic Host Configuration Protocol (DHCP) or other provisioning protocols. Today, when broadband always-on access is ubiquitous, sharing addresses over time is not considered an effective ISP-level address sharing method.

A. Network Address and Port Translation (NAPT)

A basic building block of many IPv4 address sharing mechanisms, NAPT44 (NAPT from IPv4 to IPv4, also known as Traditional NAT [8]) been deployed for a long time in home networks and enterprises, but is not commonly deployed within ISP core networks [9].

NAPT44 uses transport-layer identifiers (usually TCP and UDP ports) to multiplex privately addressed [10] hosts to a public IPv4 address. Using NAPT44, source addresses (and possibly ports) are translated as the packets are transported from an end host behind a NAPT to the Internet, and destination address (and possibly port) translation is performed in the reverse direction. When attempting to initiate a flow towards a machine behind the NAPT device the packet is sent from the Internet host to a specific address and port of the NAPT device, which appears to be the final destination. The destination address (and possibly port) are translated so that the packet is forwarded to the appropriate host (usually using RFC 1918 addressing).

NAPT44 in customer networks is understood [11], [12] although it is well known that different translators behave differently [13], even though IETF has some efforts to standardize behaviour [14], [15]. The Session Traversal Utilities for NAT (STUN) protocol was developed to enable discovery of the presence and behaviour of translators [16]. However, in ISP-level address sharing, several unforeseen technical issues arise. For example, as the NAPT44 function must reside in the ISP's core network, to address all the customers' CPEs we have to use a sufficiently large block of private (or special purpose [17]) IPv4 addresses in the access network. If this block is a private address block [10], there will be issues with overlapping address space [18]. As a NAPT44 translator is stateful, the size of its mapping table increases with the number customers [19].

B. *Related Work*

It is important to understand the difference between the inherent issues of any ISP-level address sharing and the issues related to properties of specific mechanisms. In this paper, we only discuss the latter. Ford et al. have analyzed the potential issues of IPv4 address sharing [20]. Here, we only summarize issues introduced by ISP-level address sharing that are common to all the mechanisms discussed in this paper:

- **Variable Port Requirement Dynamics.** The total number of customers able to share an IPv4 address will depend upon assumptions about each customer's average number of ports in use, and the average number of simultaneously active customers.
- **Connection to a Well-Known Port Number.** Inbound connections will not work in the general case.
- **Limited to TCP, UDP and ICMP.** All address sharing mechanisms are limited to TCP, UDP, and ICMP, thereby preventing customers from fully utilizing other transport-layer protocols of the Internet (e.g., SCTP).
- **MTU Packet Too Big Attack.** A malevolent user could send an ICMP "Packet Too Big" (Type 3, Code 4) message indicating a Next-Hop Maximum Transmission Unit (MTU) of anything down to 68 octets. This value will be cached by the off-net server for all customers sharing the address of the malevolent user. This could lead to a denial of service.
- **Traceability.** As an IPv4 address is no longer a unique identifier, tracing particular customers is challenging.
- **Reverse DNS.** Many service providers populate forward and reverse DNS zones for the public IPv4 addresses that they allocate to their customers. Where public addresses are shared across multiple customers, such strings are no longer sufficient to identify individual customers.
- **6to4 Incompatibility.** The 6to4 transition mechanism requires a publicly routable IPv4 address to function.

Huston published one of the first reviews of IPv4 address sharing mechanisms [21], where he presented CGN (NAT444, DS-Lite) and A+P approaches. He described their operation and some most important advantages and disadvantages. We extend the work presenting a mechanism classification and systematically analyzing the trade-offs and including newer address sharing mechanism proposals.

Bush et al. have presented their vision of transition to IPv6 [22], which also includes IPv4 address sharing mechanisms. They warned about consequences of deploying inappropriate mechanisms, which would result in an Internet very different from the one we know today. Furthermore, they emphasized the importance of avoiding CGNs, which make core networks too complex to easily allow for deployment of future services. Also, in their experience, it is not correct to expect that deploying more IPv4 "life support" devices will help the transition, but will delay it further. They presented a 2D space of transition mechanisms, with the first dimension being the amount of stored state and the second being the type of transition (either v4-over-v6 or v6-over-v4). In contrast, our paper focuses on IPv4 address sharing mechanisms, not on IPv6 transition mechanisms in general.

At IETF 80 Xie et al. presented comparison of address sharing mechanisms [23]. It is not clear which mechanisms are considered as the terminology is vague. They do not offer justification for some of the claims (e.g., how can customer hosts using the NAT444 be reachable from the Internet). Their comparison could benefit significantly from our classification.

In a review of recent NAT standardization efforts Wing discussed address sharing mechanisms and gave some insight into consequences of ISP-level address sharing [12]. He described Stateful NAT64 and DS-Lite and highlighted their advantages and disadvantages. However, as the emphasis of his review is not on address sharing, he did not offer a structured classification and did not give a systematic analysis of the involved trade-offs.

III. IPv4 Address Sharing Mechanisms Classification

An examination of the design space instead of individual mechanisms, allows us to determine the benefits and disadvantages of each mechanism and to see what research needs to be done to conceive new useful approaches. We are interested in features such as state storage resource required, IPv6 encouragement, requirements on the access network. We now propose five dimensions for classifying mechanisms and follow this with nine classes from the classification space.

A. *Classification Methodology*

The methodology for determining the five dimensions is as follows:

- **Mechanism Analysis**: examine all existing mechanisms and extract their properties from the IETF RFC and Internet Draft documents.
- **Form Candidate Dimensions**: group properties which describe the same aspects of mechanisms together, e.g., one mechanism might require state storage in the gateway, another may not. These are two properties of the same dimension (state storage).
- **Remove Specifics**: ignore those candidate dimensions for which at least one existing mechanism yields "Non

Applicable", e.g., the address format of stateless A+P mechanisms is not applicable in other mechanisms, where there is no address format at all.

- **Assure Unique Clustering**: where two candidate dimensions yield equal clusterings of existing mechanisms choose one using operational relevance. If there are important issues with one or more properties of the left out candidate dimension, we still discuss them.
- **Remove Less Relevant Dimension Candidates**: the final set of dimensions is refined by removing dimensions containing operationally unimportant properties. This is the most subjective step. After mechanism analysis identify important issues that were explicitly stated as such in the documents or were given as a motivation for defining one or more mechanisms. As a final check, make sure that removal one of the candidate dimensions would not lead to such an important issue being ignored.

The rest of the paper is aligned with the classification dimensions. However, as the classification is inferred from existing mechanisms, the coverage may not be complete. Nevertheless, in the following sections, we argue completeness of each individual dimension.

B. *Classification Dimensions*

1) Dimension 1: Location of the IP address sharing function: The IP address sharing function can either be located either in the *CPE* (A+P mechanisms), in the *gateway* or in the *CPE and gateway* (CGN mechanisms). In A+P case, the customer can choose between using a CPE with a port-restricted NAPT function to connect their hosts or connecting a single A+P-capable host directly to the access network. In the former case the user is in control of the translation (e.g., port-forwarding). Where there is address sharing in the gateway (CGN) it becomes a critical function of the ISP, which in turn has to manage any gateway-located NAPT function.

This dimension is important as A+P mechanisms preserve the Internet's end-to-end principle to customer premises.

Given our assumed CPE-Gateway topology described earlier and our wish to support unmodified end hosts, the address sharing function cannot be placed anywhere other than the CPE or the gateway.

2) Dimension 2: State storage in the gateway: State information in the gateway may need to be held *per flow*, *per allocation* or it can be *stateless*. Note as stateful CPE devices have been widely deployed without major difficulties, this dimension only considers the gateway, which normally is supposed to hold state for a large number of customers.

From the multiple perspectives of performance, maintenance, scalability, cost, and complexity, one of the most desired properties of a mechanism is statelessness. The process of packet traversal through the mechanism is as determined from the packet IP header [24].

Per-allocation (of port and/or address) stateful mechanisms require gateway devices store information mapping IPv4 addresses and port-sets to tunnel ID, IPv6 prefix or CPE address.

Per-flow (UDP, TCP and ICMP effectively) stateful mechanisms require one entry in the gateway state table per flow. As flows are short-lived and each customer can establish many simultaneously, this state has a high churn rate.

This dimension is important, because the volume of state to be stored influences the state synchronization, logging, processing and storage requirements of the gateway. State storage will not be more fine-grained than per-flow. On the other hand, the granularity between per-flow and per-allocation can be arbitrary (and is equivalent when allocation includes only one port). The situation where a port-set is allocated among multiple customers within one allocation does not make sense as there is no way to know to which customer each packet should be forwarded.

3) Dimension 3: Traversal method through the access network: Here we refer to means by which the payload of IPv4 packets is exchanged between the CPE and the gateway. We consider the method and extent in which packet header manipulations are required. We identify the following methods *routing*, *tunneling*, *double address family translation*, and *reversible header translation*.

Routing is the simplest traversal method. No packet header manipulation occurs and therefore IPv4 and IPv6 packets can be carried from source to destination through native networks.

By Tunneling we refer to any process of encapsulating, transporting and decapsulating a packet. For example, wrapping IPv4 packets in an additional IP header, meaning original packets travel through a non-native network intact.

Double address family translation leverages the Stateless IP/ICMP Translation Algorithm (from here on abbreviated as stateless NAT64) [25], which is a method of translating the IP header of a packet from IPv4 to IPv6 and vice versa. As the translation can be done algorithmically it is useful as a means to transport the payload originally placed into an IPv4 header over an IPv6-only access network; then translating it back to IPv4 and forwarding it to the IPv4 Internet. In this case, we need to perform the address family translation twice—in the CPE (v4 to v6) and the gateway (v6 to v4).

Reversible header translation can be seen as a special case of double address family translation, with most of the IPv4 header information preserved [26].

The trade-offs when choosing a traversal method are significant and described in the Detailed Property Analysis below. Completeness of this dimension is hard to argue as one can envision an improved tunneling or translation mechanism, which will introduce new issues for analysis. If a new method is later invented, this dimension must be extended.

4) Dimension 4: Level of IPv6 requirement: Not all IPv4 address sharing mechanisms are IPv6 transition mechanisms. Some of them require IPv6 in one or more parts of the network while others work fine without IPv6. The level of IPv6 requirement is directly related to the semantics of the translation function for address sharing. We can distinguish three cases for IPv6: *no IPv6 required*, *IPv6 partly required* and *IPv6 required*. The first case covers those mechanisms where IPv6 networking is independent of the address sharing mechanism and can optionally be provided using a traditional dual stack method. In most mechanisms IPv6 is *partly required* which means access network has to be IPv6-enabled for successful operation. Finally, some mechanisms require IPv6-

enabled customer networks, which allows for IPv6-only ISP networks, where IPv4 is present solely at the Internet border.

When considering the wide-spread adoption of IPv6, it is important to evaluate to what extent a specific mechanism encourages, supports, utilizes, or requires IPv6 in the ISP.

There are three networks considered in the assumed topology: the IPv4 Internet, the access network and the customer network. There are four possible combinations of IPv4 and IPv6 values for access and customer network. This dimension excludes the combination where IPv6 is required in the customer network and IPv6 is *not* required in the access network. Although such a mechanism could be envisioned in theory, it does not make sense as migrating customer networks to IPv6 is considered more challenging than migrating the access network as this is owned by the ISP.

5) Dimension 5: IPv4 address and port allocation policy: A mechanism either provides *static and dynamic* allocation or *static-only* allocation. In *Dynamic allocation* a port and possibly also the shared IPv4 address are selected as required by the NAPT function. These are chosen on a per-flow basis as each new flow is established. The port number and address associations may be freed and reused as the flow times out. With *Static allocation* an IPv4 address and a port-set are reserved per allocation and are then (until possible reallocation) used by NAPT function for only one customer.

In CGN static or dynamic allocation may be used. In A+P, only static allocation is possible as the address sharing function is located at the edge of the network. The CPE chooses ports for new flows from its pre-allocated set.

Note that in this context the terms static and dynamic describe the granularity and persistence of address and port allocations. They do not describe the state storage needed in the gateway. We use the terms *stateless* and *per-allocation* stateful for that (Dimension 2). Address and port allocation policy is important because it influences address sharing ratio, state storage in the gateway, and security. Both dynamic and static are the only viable options for allocation policy. The question whether or not a class of mechanism can provide *only* dynamic allocation is irrelevant, as the *static and dynamic* option denotes what policies a class of mechanism *can* support (as opposed to *must* support).

C. Review and Classification of IPv4 Address Sharing Mechanisms

In this section we identify, and describe nine classes of IPv4 address sharing mechanisms (no specific order). Table I summarizes the properties for each class. Also, we provide a per-class outgoing packet flow diagram, which demonstrates the address sharing operation. Some of the classes contain multiple mechanisms, while others only have one member. This is because some classes contain competing mechanism proposals which are still being decided on in the IETF. Some other combinations do not make sense, e.g., having an address sharing function in the CPE and dynamic address and port allocation together. Some of them warrant future study.

The flow diagrams do not show the process of provisioning a CPE. The access network interface(s) of a CPE can be configured and provisioned using one of a variety of protocols, e.g., DHCP, DHCPv6, Port Control Protocol (PCP) [27], Technical Report 069 (TR69) [28] or manually. For some scenarios, the CPE only requires an IPv4 or IPv6 prefix or both. For others, one or more port-sets or encapsulation parameters. How the CPE is provisioned with prefixes and port-sets is not important to our classification, as it does not affect tunneling, encapsulation, translation, etc. Each different form of provisioning offers a different set of features and a different level of complexity.

We now describe the basic operation of each class (figures 2 to 9). An end host sends IPv4 packets destined to the IPv4 Internet to the LAN default gateway which will be the CPE (In one of the classes, a preamble must be performed first to obtain a reachable IP address) Next, packets are forwarded by the CPE's external interface to the access network's default gateway where further processing takes place as necessary. From there, packets are forwarded to the IPv4 Internet. The numbers in the figures correspond to the consecutive steps required for sending a packet.

1) Class 1: Given that NAPT44 functionality is already present in most CPE, Class 1 mechanisms add an additional level of NAPT44 in the core of the ISP's network. ISPs have deployed such technology for a long time, because it is simple and they builds solely on well-known NAPT44 translation. This is popular for aggressive IPv4 address sharing, but the end-to-end principle of the Internet is not preserved. To reduce addressing conflicts with RFC 1918 address space, IANA has allocated a special IPv4 address block to be used by ISPs for address sharing purposes [17]. NAT444 (sometimes called double NAT or CGN) is representative of this class (Figure 2). The IETF made some effort to standardize this [29] but the Draft expired. However, another Internet Draft [30] defines required behaviour of CGNs in general.

2) Class 2: The aim here is to remove double NAT by moving the NAPT44 function to the network core and away from the CPE. IPv4 traffic is tunneled between the CPE and the gateway over an IPv6 access network, which also allows elimination of addressing conflicts between customers. DS-Lite [31] is representative of this class (Figure 3).

3) Class 3: This class is similar to Class 2 of CGN mechanisms. The main difference being allowance for other tunneling techniques rather than v4-over-v6, e.g, Point-to-Point Protocol (PPP) or Point-to-Point Over Ethernet (PPPoE). This means a Class 3 mechanism can be deployed without IPv6 at all. Gateway Initiated DS-Lite [32] is representative of this class (Figure 4).

4) Class 4: This is a class of CGN mechanisms which use IPv6 as the fundamental protocol of the access network and carry IP packet contents in IPv6 packets before translating them for forwarding over the IPv4 Internet. In order to obtain the reachable IP address of the destination host, the IPv6-only end-host first queries its DNS resolver, usually the provider's DNS64 server. The DNS64 [33] server tries to fetch an AAAA resource record for the domain in question. If the domain is not IPv6-ready, this request fails and the DNS64 server retries the query, this time by looking for an A record. Note A and AAAA record queries can be performed simultaneously to reduce

TABLE I
IPv4 ADDRESS SHARING MECHANISM CLASSES

Dimension	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Location of the IP address sharing function	CPE and gateway	gateway	gateway	gateway	CPE	CPE	CPE	CPE	gateway
State storage in the gateway	per flow	per flow	per flow	per flow	stateless	per allocation	stateless	stateless	per flow
Traversal method through the access network	routing	tunneling	tunneling	routing	tunneling	tunneling	double address family translation	reversible header translation	double address family translation
Level of IPv6 requirement	no IPv6 required	IPv6 partly required	no IPv6 required	IPv6 required	IPv6 partly required	IPv6 partly required	IPv6 partly required	IPv6 partly required	IPv6 partly required
IPv4 address and port allocation policy	static and dynamic	static and dynamic	static and dynamic	static and dynamic	static-only	static-only	static-only	static-only	static and dynamic

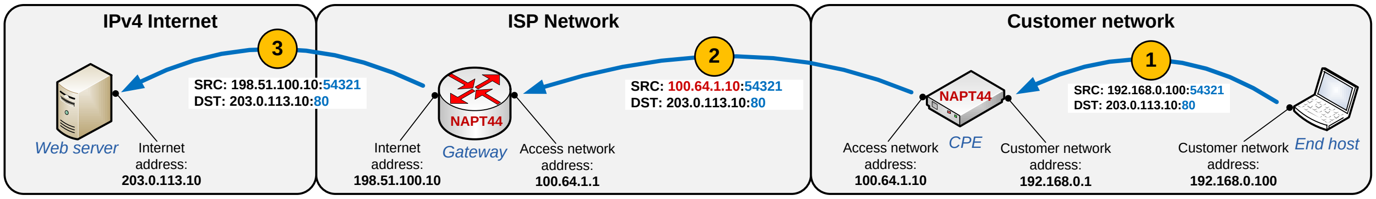

Fig. 2. In Class 1 mechanisms, IPv4 traffic is processed by two successive NAPT44 functions, in the CPE and in the gateway.

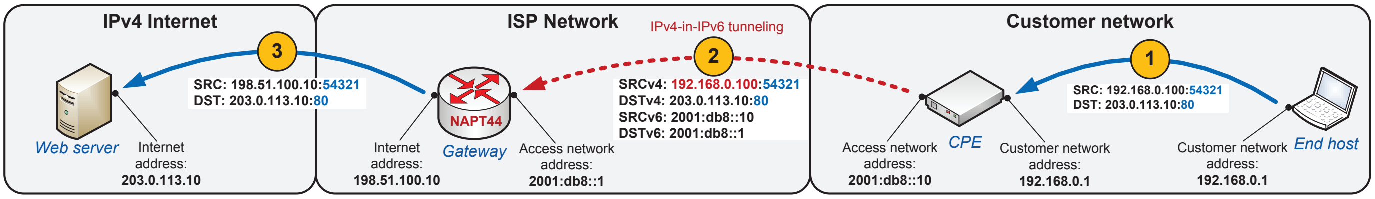

Fig. 3. In Class 2 mechanisms, IPv4 traffic is tunneled in IPv6 packets and routed to the gateway, where the NAPT44 function is located.

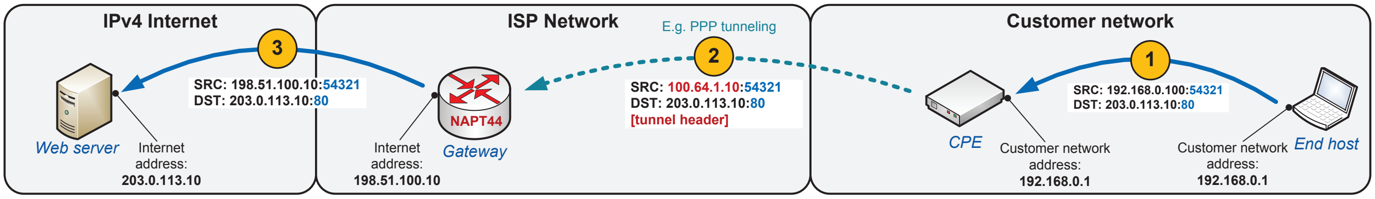

Fig. 4. In Class 3 mechanisms, IPv4 traffic is tunneled to the intermediary gateway using one of the tunneling technologies (e.g., PPP or PPPoE) and then to the border gateway, where the NAPT44 function is located.

delay. If the AAAA record exists, then the communication continues over IPv6 as usual. If no AAAA record is found, but an A record exists, the corresponding IPv4 address will be sent to the DNS64 server, which in turn algorithmically generates a synthetic IPv6 address using a common NAT64 prefix, which is routed via the NAT64 gateway. Such mechanisms do not allow direct IPv4 addressing of the end hosts, but use Network Address and Port Translation from IPv6 to IPv4 (NAPT64) in the gateway to achieve IPv4 address sharing. NAPT64 translates IPv6 packets to IPv4 packets and vice versa. This is significantly more complex than NAPT44. It causes additional issues compared to NAPT44 due to the address family translation [34]. Stateful NAT64 [35] is representative of this class (Figure 5).

5) *Class 5:* These mechanisms employ A+P at the CPE and leverage *stateless tunneling* (dimensions 2 and 3) for transferring IPv4 traffic across IPv6-only networks. All proposals in this class require no per-flow and per-allocation state in ISP's gateway. Thus, all information, required for routing packets on ISP's gateway is derived algorithmically from fixed preconfigured domain-wide settings and information encoded in IPv6 addresses. Also, as the NAPT function is located in the CPE, gateways can be lightweight. As an example figure 6 shows the addressing format and port-set encoding of 4rd. However, different encodings are also possible. The following mechanisms are representatives of this class (Figure 6): I-D.ietf-softwire-map [36], I-D.murakami-softwire-4rd [37], I-D.sun-softwire-stateless-4over6 [38], I-D.matsuhira-sa46t-as [39].

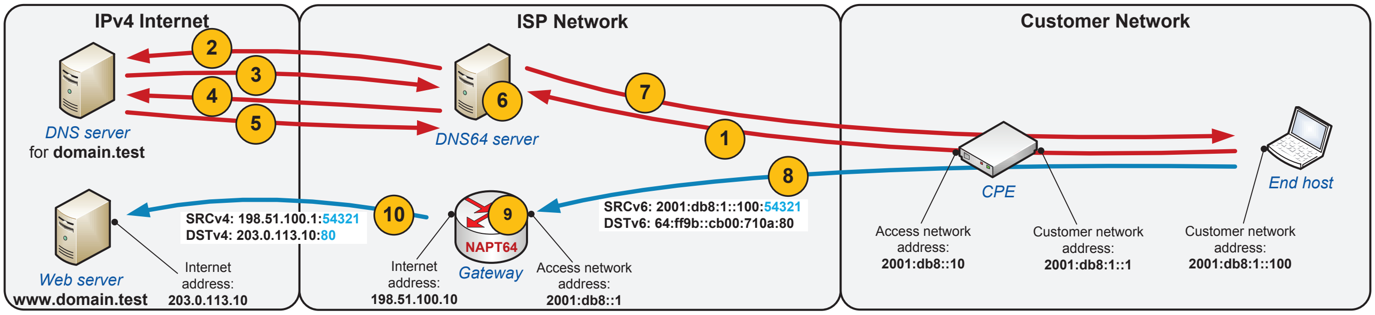

Fig. 5. In Class 4 mechanisms, DNS64 server is used by IPv6-only hosts to provide synthetic IPv6 addresses which represent IPv4 hosts.

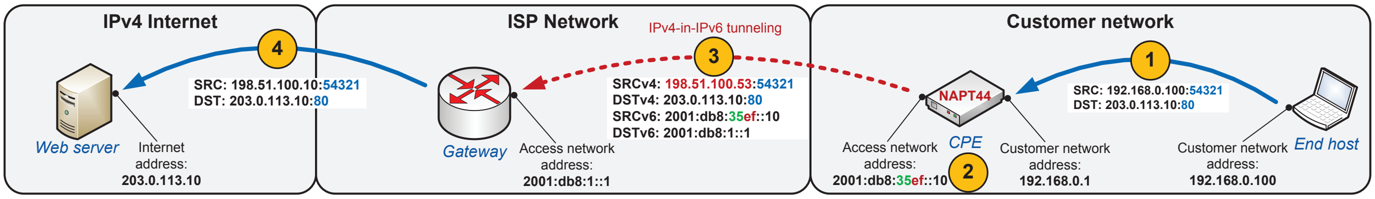

Fig. 6. In Class 5 mechanisms, IPv4 traffic is first processed by NAPT44 in the CPE and then statelessly tunneled to the gateway, which routes it to the Internet.

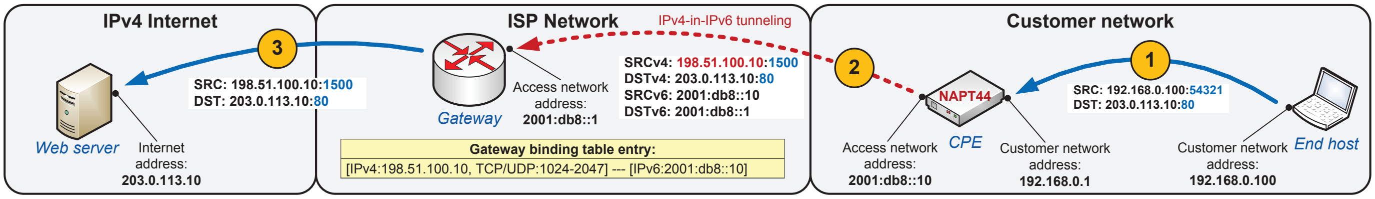

Fig. 7. Class 6 mechanisms are very similar to Class 5 mechanisms, except that they do not encode IPv4 address and port-set information in IPv6 addresses but uses a binding table in the gateway instead.

6) Class 6: Class 6 mechanisms again employ A+P, but use *stateful tunneling* as a traversal method (dimensions 2 and 3), which refers to per-allocation state required in the gateway to perform IPv4-in-IPv6 tunneling between the CPEs and the gateway. We are not referring to per-flow state, required for maintaining a NAT table in the gateway, as this is one of the A+P approaches. Additional signaling is needed to notify CPEs of their respective IPv4 addresses and port-sets: DHCP [40], PCP [41] and TR69 variants are example protocols that serve this purpose. The following mechanisms are representatives of this class (Figure 7): I-D.cui-softwire-b4-translated-ds-lite [42], I-D.zhou-softwire-b4-nat [43], I-D.draft-penno-softwire-sdnat [44].

7) Class 7: This class of A+P mechanisms is similar to Class 6 with tunneling replaced by double address family translation. This eliminates issues of tunneling. First, the packets are translated from IPv4 to IPv6 and then back from IPv6 to IPv4. Both translations are performed algorithmically and are completely stateless. In Figure 8, the addressing format and port-set encoding of Double IVI (dIVI) [45] is shown, but different formats are possible. The following mechanisms are representatives of this class (Figure 8): I-D.ietf-softwire-map-t [46], I-D.xli-behave-divi-pd [47], I-D.murakami-softwire-4v6-translation [48].

8) Class 8: This class of A+P mechanisms is similar to Class 7 with the exception of traversal method used. Reversible header translation is defined by 4rd mechanism. It uses an IPv6 fragmentation header to store some information from IPv4 header, making it reversible and almost lossless (only IPv4 options are lost, which is acceptable since they are not often used today in the Internet [49]). As this traversal method removes several limitations of tunneling and double address family translation (discussed in Detailed Property Analysis section), this mechanism is considered as a class of its own. 4rd is representative of this class [26] (Figure 8).

9) Class 9: This class of CGN mechanisms is similar to Class 2 with the exception of traversal method and translation function used. However, it was developed to provide limited (outbound, client-server) IPv4 access to IPv4-only applications on directly connected IPv6-only provisioned end-hosts (no CPE involved). 464XLAT [50] is representative of this class (Figure 9).

IV. Detailed Property Analysis

To identify mechanism trade-offs, we have to understand their properties. We discuss the issues of each property along the five dimensions of our classification.

A. Dimension 1: Location of the IP address sharing function

If the address sharing function is located in the gateway, we call such mechanisms CGNs, otherwise we refer to them as

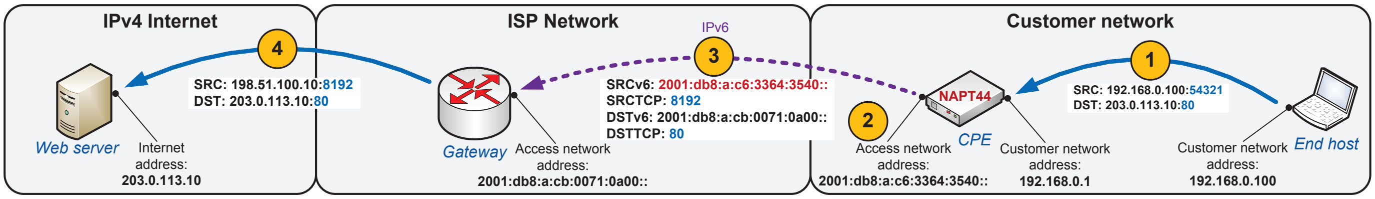

Fig. 8. In Class 7 mechanisms, IPv4 traffic is translated to IPv6 in the CPE and then back to IPv4 in the ISP's gateway.

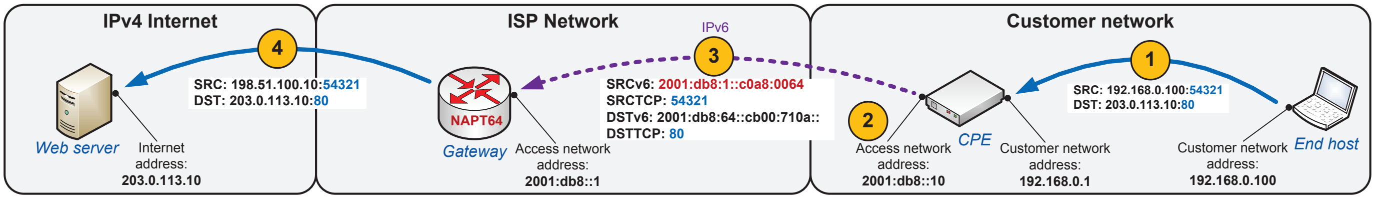

Fig. 9. In Class 9 mechanisms, IPv4 traffic is statelessly translated to IPv6 in the CPE and then statefully back to IPv4 in the ISP's gateway.

A+P mechanisms. The difference impacts support for end-to-end connectivity, gateway and CPE complexity etc.

1) CPE and gateway:

a) Port forwarding through two levels of NAPT: End-to-end connectivity is difficult to achieve, as it is non-trivial for an end host to have ports forwarded to their CPE. Existing port mapping protocols, Universal Plug and Play (UPnP) [51] and NAT Port Mapping Protocol (NAT-PMP) [52] do not support double NAPT. However, PCP [27] will support it according to the charter of IETF *pcp* working group. Unfortunately, in 2012 the IETF are still discussing PCP and it is yet to be implemented by vendors. Also, it adds additional complexity into the address sharing mechanism. If port forwarding support is not provided, applications which rely on it (e.g., BitTorrent), will not function optimally.

2) Gateway:

a) Limited control over the NAPT function: In CGN schemes, customers may not modify the NAPT44 function, e.g., adapt it to new protocols, since it is locked in the ISP's core. Installing and enabling new Application Layer Gateways (ALGs) for custom applications may invoke lawyers. Similarly, letting customers configure static port forwarding rules in the centralized NAPT44 function is impractical from an ISP's perspective and raises security considerations (denial of service, lack of authentication and confidentiality, off-path source spoofing and other threats [27], section 18). Some CGN implementations may support dynamic request of port forwarding rules by using signaling protocols such as PCP [27], NAT-PMP [52], and UPnP [51]. The latter are less adapted for CGN scenarios as the port reservation dialogue may not be successful if most of the ports are already in use by other customers.

b) Higher gateway complexity: CGN gateways are more complex because they must store and synchronize a lot of flow state (see the "Stateful per flow" discussion below). It also concentrates failure points.

3) CPE:

a) Only static IPv4 address and port-set allocation possible: A+P CPEs must be given an IPv4 address and port-set in advance, i.e., statically. If an end host does not have active flows, its ports are unused yet they can not be used by another customer.

b) Higher CPE complexity: A CPE with NAPT is more complex. As today the NAPT function is ubiquitous, that in itself is not the main issue—the problem is assuring that A+P CPE is aware of its allocated IPv4 address and port-set. Especially in per-allocation stateful (Dimension 2) mechanisms, additional signaling is required. Also, A+P CPE's parameters must be synchronized with the gateway.

B. Dimension 2: State storage in the gateway

This dimension impacts logging and high availability requirements, scalability, and address usage efficiency.

1) Per flow:

a) State synchronization: When gateways are clustered, either for high-availability or load balancing, any state storage adds significantly to the complexity of the cluster [53]. All cluster nodes must synchronize state, which is hard when state is changing rapidly. E.g, if a customer establishes a TCP flow, its entry is stored in the gateways' state tables. This must be immediately synchronized with other nodes in order for them to match any subsequent packets from the customer to this specific flow. The problem of high resource usage must also be addressed by carefully designing such clusters for traffic bursts.

b) Hairpinning: Hairpinning is when a packet is returned along the same path in the opposite direction somewhere in its way from source to destination. An IPv4 packet sourced by an end host has to be delivered to the gateway in the core network first, even it is destined to another customer of the same ISP. This is inefficient as all traffic has to be processed by the gateway. However, stateless mechanisms allow CPE-CPE direct paths.

c) Logging requirements: In many jurisdictions, ISPs are required to identify customers based on an IP address and a timestamp. Traditionally, this was feasible because every customer was assigned a unique address either dynamically (e.g., via DHCP) or statically (fixed). Even in the former case, DHCP logging was possible, as only per-allocation logging was satisfactory. However, with ISP-level address sharing it becomes harder to identify customers based solely on an IP address and a timestamp. At any moment many customers share the same IP address. When the gateway is stateful per flow, it is necessary to log all mappings of internal identifiers to public addresses. Moreover, if the authorities cannot provide ISP with the source port of the inspected connection, the ISP has to log destination IP addresses and destination port numbers, which introduces privacy concerns. Per-flow logging is resource intensive: it requires fast, reliable and large storage systems. See RFC6269 [20], section 12 for more on traceability.

d) Scalability: Scalability is critical for fast growing networks. Each new customer connected to the network causes hundreds of new flows being established. This requires larger state tables, more CPU power to match packets to the state table entries, and to synchronize clustered gateway nodes.

2) Per allocation:

a) State synchronization: Here, the entries in the gateway state table are changed when a customer is (de)allocated an IPv4 address or port-set. How fast the state changes depends on the ISP resource allocation policy and is related to the IP address sharing ratio. However, such state changes much less frequently than per-flow state.

b) Hairpinning: The issue is exactly the same as above.

c) Additional signaling: In A+P the CPE needs to know its public IPv4 address and port-set for port-restriction. The per-allocation stateful A+P mechanisms do not encode IPv4 address and port-set information into the IPv6 prefix or address. Hence, additional signaling is needed to deliver this information from the gateway to the CPE.

3) Stateless:

a) Dependency between IPv6 and IPv4 addressing: To derive the IPv4 address and the port-set from the IPv6 address or prefix assigned to the CPE, at least some bits of the IPv4 address and the port-set have to be encoded in them. If the CPE only has one IPv6 address or prefix assigned before deployment of a stateless mechanism, there are two deployment possibilities. First, complete IPv6 re-addressing in the access network can be considered, which causes service unavailability and can be operationally demanding, especially if customers already rely on static IPv6 (prefix) assignments. Second, an additional IPv6 prefix for address sharing purposes can be assigned to each CPE, which could cause routing table inflation if route aggregation is not in place. Finally, any subsequent changes in IPv4 addressing and/or port-set allocation cause IPv6 re-addressing as well.

b) Mapping rules: Mapping rules must be synchronized among all devices taking part in a stateless mechanism. These define how IPv4 prefixes reserved for IPv4 address sharing are mapped to IPv6 addresses and prefixes in the access network. If an ISP has many (smaller) IPv4 prefixes, the mapping rules can be impractical to administer.

c) Less efficient IPv4 address usage: As IPv4 address and port-set (re)allocations are non-trivial (because of the IPv4 and IPv6 addressing dependency shown above), it is more likely that ISPs will initially allocate 1024 ports to each customer, though they might not need them, rather than risk frequent reallocations as those could cause service degradation. So, in practice, stateless solutions could lead to lower IPv4 address sharing ratios than other A+P mechanisms.

d) Incompatible with discontinuous IPv4 address blocks: Also, stateless tunneling is more difficult to use when an ISP has a many smaller discontinuous IPv4 address blocks instead of a few large ones. For each IPv4 address range, separate IPv4-to-IPv6 mapping rules have to be administered in CPEs and gateways.

C. Dimension 3: Traversal method through the access network

We consider this dimension because the traversal method of a mechanism influences possible MTU issues, packet inspection issues, security and performance issues etc.

1) Routing:

a) IPv4 routing does not encourage IPv6: Ideally, IPv4 address sharing mechanisms should encourage transition to IPv6 at least in some parts of the network. However, IPv4 routing does not encourage transition of IPv4-only networks to IPv6. Of course, dual-stack can be used in this case, but, as its deployment is completely independent of such IPv4-only mechanisms, it is expected that a significant number of ISPs (short-visioned) will not consider it.

2) Tunneling:

a) Maximum Transmission Unit (MTU) issues: Different sizes of IPv4 and IPv6 headers cause problems with handling the maximum packet size to any system connecting the two address families. There are four mechanisms for dealing with this issue: Path MTU Discovery (PMTUD) [54], fragmentation [31], transport-layer negotiation such as the TCP Maximum Segment Size (MSS) option [55] and increasing MTU size on all the links in the access network at least by 40 bytes to accommodate both the IPv6 encapsulation header and the IPv4 datagram without fragmenting the IPv6 packet.

b) Packet inspection issues: Any middleboxes in the access network that process IPv4 packets have to be able to unwrap tunneling to inspect one header deeper to discover the payload properly. Examples are Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) devices, which perform deep packet inspection or special environments, e.g., some 3rd Generation Partnership Project (3GPP) and PacketCable environments or transparent web proxy caches. In these environments, significant additional support is needed in various devices [56].

c) Packet size overhead: Because of the additional header, tunneling causes bandwidth overhead compared to other traversal methods. With average payload of ≈ 550 bytes, tunneling causes around 4% overhead, while with average payload of 1400 bytes, it causes around 2% overhead [56].

d) Routing loop vulnerabilities: Tunneling makes routing loop attacks possible [57]. This vulnerability can be abused as a vehicle for traffic amplification to facilitate Denial of Service (DoS) attacks [58]. However, with address sharing mechanisms, filtering makes it relatively easy to mitigate such attacks.

3) Double address family translation:

a) Maximum Transmission Unit (MTU) issues: This issue is exactly the same as above.

b) Checksum recalculation: When the packets translated between IPv4 and IPv6, the transport-layer protocol checksums must be recalculated. This may impose a significant impact on overall performance, as whole packets have to be included in checksum recalculation. Even though stateless NAT64 avoids checksum recalculation in cases of checksum-neutral prefixes, this is not applicable to some mechanisms, where IPv6 addresses also encode port information [59].

c) Potentially limited transparency to IPv4 Do not Fragment (DF) bit: In general, stateless NAT64 is transparent to the IPv4 DF bit. However, if a stateless NAT64 implementation chooses to "provide a configuration function, that allows the translator not to include the Fragment Header for the non-fragmented IPv6 packets", which is allowed by RFC 6145 [25], end-to-end DF bit transparency is broken.

d) Potentially limited transparency to IPv4 Type Of Service (TOS) field: In general, stateless NAT64 is transparent to the IPv4 TOS octet. However, as RFC 6145 [25] states, "an implementation of a translator SHOULD support an administratively configurable option to ignore the IPv4 TOS and always set the IPv6 traffic class (TC) to zero". In this case, IPv4 TOS transparency is broken.

e) Potentially unsupported fragmented zero-checksum UDP packets: RFC 6145 [25] states that fragmented IPv4 UDP packets that do not contain a UDP checksum are not in general translated by the stateless NAT64 translator. However, this need not be the case, as the translator can be configured to forward the packet without a UDP checksum, which will also work for zero-checksum UDP packets.

f) Limited transparency to ICMP: RFC 6145 [25] defines, that some of ICMP [60] message types (13, 14, 15, 16 and others) are not translated by stateless NAT64, which means that end-to-end ICMP transparency is not preserved.

g) Loss of IPv4 header options: IP/ICMP protocol translation algorithms do not support translating IPv4 header options, which means they will be lost when a packet traverses v4-v6-v4 stateless translators. This should not have significant consequences, as IPv4 header options are very rarely used today, And even when used, approximately half of such packets are dropped somewhere on their path [49].

4) Reversible header translation:

a) Maximum Transmission Unit (MTU) issues: This issue is exactly the same as above.

Loss of IPv4 header options: As with double address family translation, reversible header translation lacks support for translating IPv4 header options.

D. Dimension 4: Level of IPv6 requirement

The level of IPv6 requirement of mechanisms will impact the future Internet and the duration of IPv4/IPv6 coexistence.

1) No IPv6 required:

a) No IPv6 encouragement: None of these mechanisms will contribute to encouraging IPv6 transition, because operators are not required by any means to even consider deploying IPv6 in any of their networks.

b) Administration of IPv4 infrastructure: We consider IPv4 protocol a legacy protocol, which means that eventually, it will fade away and at that time administrating IPv4 infrastructure will not be necessary any more. Assuming IPv6 deployment is in place, IPv4 administration contributes extra significant network administration cost.

2) IPv6 partly required:

a) IPv4 in customer networks: Since access networks normally represent a large part of an ISP's network, migrating them to IPv6 is a substantial move in the direction of IPv6 transition. However, if customer networks remain IPv4-only (or even dual-stack), this means IPv4 will be kept in use for a long time, which will prolong the transition to IPv6-only Internet. In this aspect, IPv4 address scarcity can be seen as a strong driver towards IPv6-only networks where feasible.

b) Administration of IPv4 infrastructure: The is exactly the same as for mechanisms where no IPv6 is required.

3) IPv6 required:

a) IPv4-only application incompatibility: We expect that at some point, the ISPs who find it difficult to administer IPv6 *and* IPv4 addressing in customer networks, will consider deploying mechanisms which allow for IPv6-only customer networks. On IPv6-only end hosts IPv6 applications without support for IPv6 will not work. [34].

b) IP protocol-aware application incompatibility: Because connection endpoints use different address families, NAPT64 introduces incompatibilities with some application layer protocols as shown in [34]. This is true for IP-protocol-aware application protocols, BitTorrent, FTP, and Session Initiation Protocol (SIP) [61] being widely-used examples. For every such protocol an ALG can be constructed, but each new ALG contributes more complexity to network operation.

c) Only IPv6-enabled hosts supported: Public IPv4 address sharing among dual-stack and IPv4-only end hosts is not supported by such mechanisms. This means that any non-IPv6 ready devices will not be able to connect to IPv4 services, which can be a serious limitation in heterogeneous environments. Legacy devices such as old faxes or printers with embedded networking are problematic examples.

d) Requires DNS64 service for operation: These mechanisms require DNS64 in order to be effective, this means another service to administer. It also means IPv4 traffic destined to IPv4 address literals are not supported. This means, that if the end host tries to browse to http://203.0.110.10, requests will fail immediately, as no DNS request is made to cause synthesis of a usable IPv6 address.

E. Dimension 5: IPv4 address and port allocation policy

This dimension is important as it impacts address sharing ratio, state storage in the gateway and security.

1) Static and dynamic: Although mechanisms with this property support both allocation policies, we discuss the issue with dynamic allocation in this section and issues with static allocation in the next section.

Stateful per flow: Dynamic allocation is stateful per flow, so we must record which resources are allocated to which flows. This introduces logging, scalability, state synchronization, and other issues (see Stateful per flow section above).

TABLE II
TRADE-OFFS: CLASSES OF MECHANISMS TRADING THE DESIRED FEATURES (UNORDERED)

Address-Plus-Port	**Carrier-Grade-NAT**
End-to-end connectivity (control over NAPT function)	Simple CPEs, easy provisioning and management
Scalability	Technology mature & available
Stateful	**Stateless**
Flexible addressing, no IPv6-IPv4 addressing dependency	Easy Load-Balancing
Efficient IPv4 address usage	Easy High-Availability
Scattered address-space supported	CPE-to-CPE direct paths
Tunneling	**Double Translation**
Keeps IPv4 packets intact	No packet inspection issues
No checksum recalc. overhead	No tunneling packet size overhead
Mature and widespread method	No routing loop vulnerability
IPv6 Required	**IPv6 Not Required**
Transition encouraged	Easy deployment
Less administration	Legacy application compatibility
Static Allocation	**Dynamic Allocation**
Manageable state	High address sharing ratio
Efficient logging	More secure

2) Static-only:

a) Low address sharing ratio: Since port-sets are allocated to customers instead of individual ports to flows, many ports remain unused. This is due to the need to allocate a large enough a port-set to a customer so that they will never use all of the allocated ports (which would cause service degradation). And because the number of used ports by a customer can vary significantly, the worst case becomes the universal case. This means that, for any given address space, fewer customers can be offered service than with dynamic allocation.

b) Port randomization security issues: The TCP protocol is inherently vulnerable to spoofed off-path packet injection attacks [62]. To implement an attack on a TCP session established between two hosts, the adversary must guess the 4-tuple (source port, destination port, source address, destination address) of the TCP connection together with 32-bit sequence ID. The attack is feasible and static port allocation makes the problem even worse—the 16-bit port space becomes smaller, which makes the 4-tuple easier to guess [56].

V. TRADE-OFF ANALYSIS

Having determined an appropriate set of dimensions for classifying address sharing mechanisms and performed a detailed property analysis of each dimension it now remains to select the most significant of these to determine the trade-offs. Table II shows a summary.

A. Carrier-Grade-NAT versus Address-Plus-Port

Infrastructure simplicity and ease of deployment together with technology maturity and availability are important features for ISPs as they easily translate to reduced costs. Also, waiting for A+P mechanisms to become widely delivered by vendors can mean losing customers in the meantime. However, ISP customers require end-to-end protocols and are not concerned with infrastructure issues. If users are not able to traverse CGNs to use very popular applications (gaming, VoIP, peer-to-peer, streaming), or if these applications show significant performance degradation, the ISP market will start to segment by the quality of NAT traversal support (through ALGs). The scalability of A+P solutions is a further cost-reducing benefit to the ISP.

B. Stateful versus Stateless

The benefits of stateful gateways mostly relate to stateful A+P solutions rather to CGN solutions. ISPs located in regions where Internet penetration is still gaining momentum often have many scattered IPv4 address ranges, which makes them good candidates for stateful solutions. Also, as these ISPs value IPv4 addresses, being able to effortlessly allocate different port-sets to customers in more nimble way is also welcome. In networks where IPv6 is already deployed in the access and the core networks, complete IPv6 re-addressing increases cost of deploying a (stateless) IPv4 address sharing mechanism. In this case, independence of IPv6 and IPv4 addressing schemes is beneficial, although a separate IPv6 addressing scheme for address sharing purposes can be used, which introduces additional administrative complexity and cost. However, stateless solutions are attractive in several scenarios. Large ISPs with significant inter-customer traffic are motivated to search for stateless solutions which allow for direct inter-customer communication. Also, avoiding state eliminates many of the difficulties brought by state synchronization requirements, including those involved in supporting high-availability and load-balancing.

C. Tunneling versus Double Translation

The third dimension has four different traversal methods. However, as routing and reversible header translation are related to specific mechanisms rather than mechanisms classes, the real decision is whether an ISP should choose a mechanism with IPv4-in-IPv6 tunneling or double stateless NAT64 translation. The former is a mature and proven method of carrying IPv4 packets over IPv6-only networks. The caveats are known, and workarounds are available. Tunneling protects the inner packet from being semantically distorted. However, double translation avoids the caveats of tunneling and also requires less processing in the path from the CPE to the gateway. This point is valid especially in those networks where packet inspection is performed.

D. IPv6 Required versus IPv6 Not Required

In our classification IPv6 Required means required in the access and customer networks. Such schemes highly encourage IPv6 transition as only the ISP border remains configured with IPv4 address(es). IPv6-only networks require one Internet protocol less to administer. However, those mechanisms which do not require any IPv6 deployment (not even in the access network) are usually easier to deploy quickly and do not cause incompatibilities with legacy IPv4-only software.

E. Static Allocation versus Dynamic Allocation

By choosing a mechanism with dynamic address and port allocation, the ISP can use a very small number of IPv4 addresses to support many customers as the sharing ratio can an order of magnitude higher than a static allocation mechanism. However, even in the static case we can easily multiplex 64 customers on one IPv4 address with each customer allocated 1024 ports. Compared to current situation where one customer is allocated one public IPv4 address, the compression of static allocation is a major benefit. Together with the reduced flow state storage of static allocation comes more efficient logging which is especially important as the ISPs frequently offer faster plans to customers, and being able to log flows in the dynamic allocation case is about four orders of magnitude more storage-intensive than static allocation, where only port-set allocations need be logged. It is important to note that it is dynamic allocation which causes logging problems, not CGN mechanisms alone. Dynamic allocation schemes are less prone to spoofed off-path injection attacks on TCP sessions.

VI. Conclusion

In this paper, we presented a novel classification of IPv4 address sharing mechanisms, which was then used to discuss and analyze their various properties. Our goal was to present the trade-offs involved in choosing a specific mechanism in an understandable but a consistent way. First, we defined an IPv4 address sharing mechanism space of five dimensions. Next, we systematically reviewed all mechanisms proposed to date, classifying them using our taxonomy. Moreover, we analyzed the issues related to the properties of mechanisms along the dimensions of our classification. Finally, we summarized the property analysis into a qualitative trade-off analysis, focusing on trading benefits of specific properties along the dimensions of the classification.which are now short of available IPv4 addresses and are forced to deploy one or more IPv4 address sharing mechanisms.

The CGN versus A+P dilemma is not the same as "NAPT-in-the-CPE" versus "NAPT-in-the-core" dilemma, which we believe is the common misperception. As our classification finally separates the dimensions and analyzes them individually, this enables us to see clearer that logging complexity, for example, is not dependent on the location of the NAPT function, but rather on the IPv4 address and port allocation policy; a very important difference.

Address translation and port-restriction can be regarded as two separate functions, which can be performed at different places independently. This opens a space for new mechanisms which have not been envisioned before.

Address family translation in the context of traversal method is completely unrelated to classical NAPT. In our experience there is a lot of misunderstanding of various roles translation (in general) can play in the context of address sharing mechanisms. This misunderstanding is mainly based on the fact that there is no fundamental framework available to the community in which to operate and view the various proposals. The chaos in the IETF is an obvious result of this confusion. A solid framework will help in more structured progress on this topic in the future.

The only actual address sharing mechanism which really pushes forward the transition to IPv6 is Stateful NAT64 (Class 4). All other (classes of) mechanisms are more tolerant to IPv4. More research is needed in this direction if our goal is to encourage IPv6 transition.

We realize the IETF is still actively working on defining details of and standardizing various IPv4 address sharing mechanisms. Although industry is pressing for a stop to the research and development of new mechanisms and for standardization and deployment of current proposals, we are confident that there are still gaps to fill in this area. For example, using our classification, one may envision a mechanism with the following properties: address sharing function located in the CPE (A+P), stateless gateway, routing as the access network traversal method, IPv6 required in the access and the customer network, and static address and port allocation. Such mechanism would highly encourage IPv6 transition and would have all the benefits of A+P and stateless mechanisms. Another idea for future work is to develop a theoretical performance evaluation model for address sharing mechanisms. This way the performance of mechanisms could be evaluated without using actual implementations. Furthermore, analyzing implications for measurement of technologies would be useful in order to develop a mechanism detection and identification model. For all of these tasks, the classification presented in this paper will provide grounds because it provides the necessary abstraction of the mechanisms in the form of various classes.

Acknowledgments

We are indebted to Simon Perreault, Ian Dickinson, Gang Chen, Ole Trøan, Dan Wing, Ibrahim Rashid and the team at Sky, as well as Ahmed Abubakar, Roshan Mosaheb, Mehdi Mousavi, and Jozef Woods at Loughborough University for their valuable feedback on this work. This work was partially financed by the European Union, European Social Fund, and by a gift from Cisco University Research Program Fund, a corporate advised fund of Silicon Valley Community Foundation, grant 2011-89493(3696).

References

[1] (2011, May) Policies for IPv4 address space management in the Asia Pacific region. Asia Pacific Network Information Centre. [Online]. Available: http://www.apnic.net/policy/add-manage-policy#9.10

[2] (2005, Jan.) IPv6 address space. Microsoft. [Online]. Available: http://technet.microsoft.com/en-us/library/cc781652(v=ws.10).aspx

[3] L. Eggert. IPv6 Deployment Trends. [Online]. Available: http://eggert.org/meter/ipv6

[4] IPv6 Statistics. Google. [Online]. Available: http://www.google.com/intl/en/ipv6/statistics/

[5] E. Nordmark and R. E. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers," RFC 4213 (Proposed Standard), Internet Engineering Task Force, Oct. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4213.txt

[6] L. J. Camp, "IPv6 Diffusion Through the Lens of Economics of Security," in *RIPE 56*, Berlin, Germany, May 5–9, 2008. [Online]. Available: http://meetings.ripe.net/ripe-56/presentations/Camp-IPv6_Economics_Security.pdf

[7] R. Bush and D. Meyer, "Some Internet Architectural Guidelines and Philosophy," RFC 3439 (Informational), Internet Engineering Task Force, Dec. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3439.txt

[8] P. Srisuresh and K. B. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022 (Informational), Internet Engineering Task Force, Jan. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3022.txt

[9] "IPv6 Industry Survey Results," Jun. 2012. [Online]. Available: http://btdiamondip.com/2012_IPv6_Survey_Report.aspx

[10] Y. Rekhter, R. G. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets," RFC 1918 (Best Current Practice), Internet Engineering Task Force, Feb. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1918.txt

[11] M. Smith and R. Hunt, "Network security using NAT and NAPT," in *10th IEEE International Conference on Networks (ICON)*, Mumbai, India, Dec. 18–21, 2002, pp. 355–360. [Online]. Available: http://dx.doi.org/10.1109/ICON.2002.1033337

[12] D. Wing, "Network Address Translation: Extending the Internet Address Space," *Internet Computing, IEEE*, vol. 14, no. 4, pp. 66–70, Aug. 2010. [Online]. Available: http://dx.doi.org/10.1109/MIC.2010.96

[13] G. Huston, "Anatomy: A Look Inside Network Address Translators," *The Internet Protocol Journal*, vol. 7, no. 3, pp. 2–32, Sep. 2004. [Online]. Available: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-3/anatomy.html

[14] F. Audet and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP," RFC 4787 (Best Current Practice), Internet Engineering Task Force, Jan. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4787.txt

[15] S. Guha, K. Biswas, B. Ford, S. Sivakumar, and P. Srisuresh, "NAT Behavioral Requirements for TCP," RFC 5382 (Best Current Practice), Internet Engineering Task Force, Oct. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5382.txt

[16] D. MacDonald and B. Lowekamp, "NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN)," RFC 5780 (Experimental), Internet Engineering Task Force, May 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5780.txt

[17] J. Weil, V. Kuarsingh, C. Donley, C. Liljenstolpe, and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space," RFC 6598 (Best Current Practice), Internet Engineering Task Force, Apr. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6598.txt

[18] P. Srisuresh and B. Ford, "Unintended Consequences of NAT Deployments with Overlapping Address Space," RFC 5684 (Informational), Internet Engineering Task Force, Feb. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5684.txt

[19] S. Alcock. (2008, Aug.) Research into the Viability of Service-Provider NAT. WAND Network Research Group, University of Waikato. [Online]. Available: http://www.wand.net.nz/~salcock/someisp/flow_counting/result_page.html

[20] M. Ford, M. Boucadair, A. Durand, P. Levis, and P. Roberts, "Issues with IP Address Sharing," RFC 6269 (Informational), Internet Engineering Task Force, Jun. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6269.txt

[21] G. Huston, "NAT++: Address Sharing in IPv4," *The Internet Protocol Journal*, vol. 13, no. 2, pp. 2–15, Jun. 2010. [Online]. Available: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_13-2/132_address.html

[22] R. Bush, M. Townsley, and D. Wing, "An IPv4 End of Life Plan: A Shared Vision for IPv6," in *Asia Pacific Regional Internet Conference on Operational Technologies (APRICOT)*, New Delhi, India, Feb. 21–Mar. 2, 2012. [Online]. Available: http://www.apricot2012.net/__data/assets/pdf_file/0016/45241/120229.apops-v4-life-extension.pdf

[23] C. Xie and Q. Sun, "Problem Statement for Operational IPv6/IPv4 Co-Existence," in *IETF 80*, Prague, Czech Republic, Mar. 27–Apr. 1, 2011. [Online]. Available: http://www.ietf.org/proceedings/80/slides/v6ops-15.pdf

[24] M. Boucadair, S. Matsushima, Y. Lee, O. Bonness, I. Borges, and G. Chen, "Motivations for Carrier-side Stateless IPv4 over IPv6 Migration Solutions," Internet-Draft (work in progress), draft-ietf-softwire-stateless-4v6-motivation-05, Internet Engineering Task Force, Nov. 2012. [Online]. Available: http://tools.ietf.org/html/draft-ietf-softwire-stateless-4v6-motivation-05

[25] X. Li, C. Bao, and F. Baker, "IP/ICMP Translation Algorithm," RFC 6145 (Proposed Standard), Internet Engineering Task Force, Apr. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6145.txt

[26] S. Jiang, R. Després, R. Penno, Y. Lee, G. Chen, and M. Chen, "IPv4 Residual Deployment via IPv6 - a Stateless Solution (4rd)," Internet-Draft (work in progress), draft-ietf-softwire-4rd-04, Internet Engineering Task Force, Oct. 2012. [Online]. Available: http://tools.ietf.org/html/draft-ietf-softwire-4rd-04

[27] D. Wing, S. Cheshire, M. Boucadair, R. Penno, and P. Selkirk, "Port Control Protocol (PCP)," Internet-Draft (work in progress), draft-ietf-pcp-base-29, Internet Engineering Task Force, Nov. 2012. [Online]. Available: http://tools.ietf.org/html/draft-ietf-pcp-base-29

[28] S. Banks, A. Colmegna, and T. Spets, "TR-069 Amendment 4: CPE WAN Management Protocol," Broadband Forum Technical Report, Jul. 2011. [Online]. Available: http://www.broadband-forum.org/technical/download/TR-069_Amendment-4.pdf

[29] I. Yamagata, Y. Shirasaki, A. Nakagawa, J. Yamaguchi, and H. Ashida, "NAT444," Internet-Draft (work in progress), draft-shirasaki-nat444-06, Internet Engineering Task Force, Jul. 2012. [Online]. Available: http://tools.ietf.org/html/draft-shirasaki-nat444-06

[30] S. Perreault, I. Yamagata, S. Miyakawa, A. Nakagawa, and H. Ashida, "Common requirements for Carrier Grade NATs (CGNs)," Internet-Draft (work in progress), draft-ietf-behave-lsn-requirements-10, Internet Engineering Task Force, Dec. 2012. [Online]. Available: http://tools.ietf.org/html/draft-ietf-behave-lsn-requirements-10

[31] A. Durand, R. Droms, J. Woodyatt, and Y. L. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion," RFC 6333 (Proposed Standard), Internet Engineering Task Force, Aug. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6333.txt

[32] F. Brockners, S. Gundavelli, S. Speicher, and D. Ward, "Gateway-Initiated Dual-Stack Lite Deployment," RFC 6674 (Proposed Standard), Internet Engineering Task Force, Jul. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6674.txt

[33] M. Bagnulo, A. Sullivan, P. Matthews, and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers," RFC 6147 (Proposed Standard), Internet Engineering Task Force, Apr. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6147.txt

[34] N. Škoberne and M. Ciglarič, "Practical Evaluation of Stateful NAT64/DNS64 Translation," *Advances in Electrical and Computer Engineering*, vol. 11, no. 3, pp. 49–54, Aug. 2011. [Online]. Available: http://dx.doi.org/10.4316/AECE.2011.03008

[35] M. Bagnulo, P. Matthews, and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers," RFC 6146 (Proposed Standard), Internet Engineering Task Force, Apr. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6146.txt

[36] O. Troan, W. Dec, X. Li, C. Bao, S. Matsushima, and T. Murakami, "Mapping of Address and Port with Encapsulation (MAP)," Internet-Draft (work in progress), draft-ietf-softwire-map-04, Internet Engineering Task Force, Feb. 2013. [Online]. Available: http://tools.ietf.org/html/draft-ietf-softwire-map-04

[37] T. Murakami, O. Troan, and S. Matsushima, "IPv4 Residual Deployment on IPv6 infrastructure - protocol specification," Internet-Draft (work in progress), draft-murakami-softwire-4rd-01, Internet Engineering Task Force, Sep. 2011. [Online]. Available: http://tools.ietf.org/html/draft-murakami-softwire-4rd-01

[38] Q. Sun, C. Xie, Y. Cui, J. Wu, P. Wu, C. Zhou, and Y. L. Lee, "Stateless 4over6 in access network," Internet-Draft (work in progress), draft-sun-softwire-stateless-4over6-00, Internet Engineering Task Force, Sep. 2011. [Online]. Available: http://tools.ietf.org/html/draft-sun-softwire-stateless-4over6-00

[39] N. Matsuhira, "Stateless Automatic IPv4 over IPv6 Tunneling with IPv4 Address Sharing," Internet-Draft (work in progress), draft-matsuhira-sa46t-as-04, Internet Engineering Task Force, Jan. 2013. [Online]. Available: http://tools.ietf.org/html/draft-matsuhira-sa46t-as-04

[40] G. Bajko, T. Savolainen, M. Boucadair, and P. Levis, "Port Restricted IP Address Assignment," Internet-Draft (work in progress), draft-bajko-pripaddrassign-04, Internet Engineering Task Force, Mar. 2012. [Online]. Available: http://tools.ietf.org/html/draft-bajko-pripaddrassign-04

[41] Q. Sun, M. Boucadair, S. S. C. Zhou, T. Tsou, and S. Perreault, "Port Control Protocol (PCP) Extension for Port Set Allocation," Internet-Draft (work in progress), draft-tsou-pcp-natcoord-10, Internet Engineering Task Force, Feb. 2013. [Online]. Available: http://tools.ietf.org/html/draft-tsou-pcp-natcoord-10

[42] Y. Cui, Q. Sun, M. Boucadair, T. Tsou, Y. L. Lee, and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture," Internet-Draft (work in progress), draft-cui-softwire-b4-translated-ds-lite-11, Internet Engineering Task Force, Feb. 2013. [Online]. Available: http://tools.ietf.org/html/draft-cui-softwire-b4-translated-ds-lite-11

[43] X. Deng, M. Boucadair, C. Zhou, T. Tsou, and G. Bajko, "NAT offload extension to Dual-Stack lite," Internet-Draft (work in progress), draft-zhou-softwire-b4-nat-04, Internet Engineering Task Force, Oct. 2011. [Online]. Available: http://tools.ietf.org/html/draft-zhou-softwire-b4-nat-04

[44] R. Penno, A. Durand, and A. Clauberg, “Stateless DS-Lite,” Internet-Draft (work in progress), draft-penno-softwire-sdnat-02, Internet Engineering Task Force, Mar. 2012. [Online]. Available: http://tools.ietf.org/html/draft-penno-softwire-sdnat-02

[45] C. Bao, X. Li, Y. Zhai, and W. Shang, “dIVI: Dual-Stateless IPv4/IPv6 Translation,” Internet-Draft (work in progress), draft-xli-behave-divi-04, Internet Engineering Task Force, Oct. 2011. [Online]. Available: http://tools.ietf.org/html/draft-xli-behave-divi-04

[46] X. Li, C. Bao, W. Dec, O. Troan, S. Matsushima, and T. Murakami, “Mapping of Address and Port using Translation (MAP-T),” Internet-Draft (work in progress), draft-ietf-softwire-map-t-01, Internet Engineering Task Force, Feb. 2013. [Online]. Available: http://tools.ietf.org/html/draft-ietf-softwire-map-t-01

[47] X. Li, C. Bao, W. Dec, R. Asati, C. Xie, and Q. Sun, “dIVI-pd: Dual-Stateless IPv4/IPv6 Translation with Prefix Delegation,” Internet-Draft (work in progress), draft-xli-behave-divi-pd-01, Internet Engineering Task Force, Sep. 2011. [Online]. Available: http://tools.ietf.org/html/draft-xli-behave-divi-pd-01

[48] T. Murakami, G. Chen, H. Deng, and W. Dec, “4via6 Stateless Translation,” Internet-Draft (work in progress), draft-murakami-softwire-4v6-translation-00, Internet Engineering Task Force, Jul. 2011. [Online]. Available: http://tools.ietf.org/html/draft-murakami-softwire-4v6-translation-00

[49] R. Fonseca, G. M. Porter, R. H. Katz, S. Shenker, and I. Stoica, “IP Options are not an option,” Electrical Engineering and Computer Sciences, University of California at Berkeley, Tech. Rep. UCB/EECS-2005-24, Dec. 2005. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2005/EECS-2005-24.pdf

[50] M. Mawatari, M. Mawashima, and C. Byrne, “464XLAT: Combination of Stateful and Stateless Translation,” Internet-Draft (work in progress), draft-ietf-v6ops-464xlat-10, Internet Engineering Task Force, Feb. 2013. [Online]. Available: http://tools.ietf.org/html/draft-ietf-v6ops-464xlat-10

[51] “Information technology – UPnP Device Architecture – Part 1: UPnP Device Architecture Version 1.0,” ISO 29341-1:2011, International Organization for Standardization, Sep. 2011. [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=57494

[52] S. Cheshire and M. Krochmal, “NAT Port Mapping Protocol (NAT-PMP),” Internet-Draft (work in progress), draft-cheshire-nat-pmp-07, Internet Engineering Task Force, Jan. 2013. [Online]. Available: http://tools.ietf.org/html/draft-cheshire-nat-pmp-07

[53] Y.-H. Feng, N.-F. Huang, R.-T. Liu, and M.-H. Wu, “Flow Digest: A State Replication Scheme for Stateful High Availability Cluster,” in *IEEE International Conference on Communications (ICC)*. Glasgow, Scotland: IEEE, Jun. 24–28, 2007, pp. 1298–1303. [Online]. Available: http://dx.doi.org/10.1109/ICC.2007.219

[54] J. Mogul and S. Deering, “Path MTU discovery,” RFC 1191 (Draft Standard), Internet Engineering Task Force, Nov. 1990. [Online]. Available: http://www.ietf.org/rfc/rfc1191.txt

[55] J. Postel, “Domain name system implementation schedule,” RFC 897, Internet Engineering Task Force, Feb. 1984, updated by RFC 921. [Online]. Available: http://www.ietf.org/rfc/rfc897.txt

[56] W. Dec, R. Asati, C. Bao, H. Deng, and M. Boucadair, “Stateless 4Via6 Address Sharing,” Internet-Draft (work in progress), draft-dec-stateless-4v6-04, Internet Engineering Task Force, Oct. 2011. [Online]. Available: http://tools.ietf.org/html/draft-dec-stateless-4v6-04

[57] G. Nakibly and F. L. Templin, “Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations,” RFC 6324 (Informational), Internet Engineering Task Force, Aug. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6324.txt

[58] M. J. Handley, E. Rescorla, and IAB, “Internet Denial-of-Service Considerations,” RFC 4732 (Informational), Internet Engineering Task Force, Dec. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4732.txt

[59] C. Bao, C. Huitema, M. Bagnulo, M. Boucadair, and X. Li, “IPv6 Addressing of IPv4/IPv6 Translators,” RFC 6052 (Proposed Standard), Internet Engineering Task Force, Oct. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc6052.txt

[60] J. Postel, “Internet Control Message Protocol,” RFC 792 (Standard), Internet Engineering Task Force, Sep. 1981, updated by RFCs 950, 4884. [Online]. Available: http://www.ietf.org/rfc/rfc792.txt

[61] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “SIP: Session Initiation Protocol,” RFC 3261 (Proposed Standard), Internet Engineering Task Force, Jun. 2002, updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[62] A. Ramaiah, R. Stewart, and M. Dalal, “Improving TCP’s Robustness to Blind In-Window Attacks,” RFC 5961 (Proposed Standard), Internet Engineering Task Force, Aug. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5961.txt

Nejc Škoberne is a junior researcher at Viris and studying for a Ph.D. at Faculty of Computer and Information Science, University of Ljubljana. Nejc has more than 10 years of experience in system/network administration and information security industry. He finished his B.Sc. - Thesis title Improvement of the pfSense firewall with user services. His research focus is on IPv6 transition and coexistence.

Olaf Maennel is a lecturer at Loughborough University in the United Kingdom since September 2009. Before that he was at Deutsche Telekom Laboratories and at the School of Mathematical Science at the University of Adelaide in South Australia. He received his Ph.D. (Dr. rer. net) from the Technical University in Munich (Germany) in 2005. His research interests are routing, network security, active measurements, next generation internet technology, as well as configuration management.

Iain Phillips is a Senior Lecturer at Loughborough University in the UK. He received B.Sc. and Ph.D. degrees from Computer Science, Manchester University, UK. He has worked at Loughborough since 1992 in both EE and CS departments, being Head of CS 2008-11. From 2012-14 he will be Chair of the Council of Professors and Heads of Computing (CPHC). His research interests include network architectures considering performance and algorithms for the Internet and Wireless Sensor Networks.

Randy Bush is a Research Fellow and Network Operator at Internet Initiative Japan, Japan’s first commercial ISP, and a visiting Professor at Loughborough University. He specializes in network measurement and security, routing protocols, and IPv6 deployment. He is also a lead designer of the BGP security effort. Randy has been in computing for over 45 years, and has a few decades of Internet operations experience. He was the engineering founder of Verio, now NTT/Verio. He has been heavily involved in transferring Internet technologies to developing economies.

Jan Žorž started his professional career in RS-232/VAX VMS world in 1992 and continued through Novell and Windows environments all the way to Solaris and other UNIX derivatives. For the last seven years Jan has been a consultant in the IT field, specializing in IPv6. He co-founded the not-for-profit Go6 institute, a Slovenian IPv6 initiative whose main objective is to raise IPv6 awareness in Slovenia and alert the community to the fact that we are approaching extensive changes on the Internet.

Mojca Ciglarič is a head of Computer Communications Laboratory at Faculty of Computer and information Science, University of Ljubljana, Slovenia, and research director at Cloud Security Alliance, Slovenian chapter. She holds a PhD in Computer Science since 2003 and works as assistant professor at University of Ljubljana since 2006. Her research interests include computer networks and distributed systems as well as network security.