

A Multi-perspective Analysis of Carrier-Grade NAT Deployment

Philipp Richter¹, Florian Wohlfart², Narseo Vallina-Rodriguez³,
Mark Allman³, Randy Bush⁵, Anja Feldmann¹, Christian Kreibich^{3,6},
Nicholas Weaver³, Vern Paxson^{3,4}

¹TU Berlin, ²TU München, ³ICSI, ⁴UC Berkeley, ⁵Internet Initiative Japan, ⁶Lastline

ABSTRACT

As ISPs face IPv4 address scarcity they increasingly turn to network address translation (NAT) to accommodate the address needs of their customers. Recently, ISPs have moved beyond employing NATs only directly at individual customers and instead begun deploying *Carrier-Grade NATs* (CGNs) to apply address translation to many independent and disparate endpoints spanning physical locations, a phenomenon that so far has received little in the way of empirical assessment. In this work we present a broad and systematic study of the deployment and behavior of these middle-boxes. We develop a methodology to detect the existence of hosts behind CGNs by extracting non-routable IP addresses from peer lists we obtain by crawling the BitTorrent DHT. We complement this approach with improvements to our Netylizr troubleshooting service, enabling us to determine a range of indicators of CGN presence as well as detailed insights into key properties of CGNs. Combining the two data sources we illustrate the scope of CGN deployment on today's Internet, and report on characteristics of commonly deployed CGNs and their effect on end users.

1. INTRODUCTION

As originally designed, the Internet architecture calls for IP addresses to uniquely identify devices. This structure lays the foundation for a peer-to-peer system that facilitates direct communication between hosts. However, this model runs into trouble once addresses become scarce. This situation first manifested in home networks, where ISPs provided subscribers with only a single IP address while the number of devices in home networks ballooned.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IMC 2016, November 14 - 16, 2016, Santa Monica, CA, USA

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-4526-2/16/11...\$15.00

DOI: <http://dx.doi.org/10.1145/2987443.2987474>

Today, Network Address Translation (NAT) [18] is ubiquitous at the edge of home networks to meet both the ISPs' desire to conserve IP addresses and the users' requirement of connecting a multitude of devices. IP address scarcity has long moved beyond home networks and onto the global stage [36]. Thus, even though IPv6 is gaining momentum [14], large ISPs are confronted by address shortages, as we illustrate in § 2, and hence turn to a well-worn coping technique: NAT.

Instead of aggregating small populations of tightly-knit users and devices within one residence under a single IP address, *Carrier-Grade NATs* (CGNs) apply NAT to many independent and disparate endpoints spanning physical locations. On one level we can view CGNs as representing a second instantiation of a well-known technique for combating address shortages. While tempting, conflating CGNs with small edge-based NATs represents a false equivalence, for two reasons: (i) by operating at large scales, CGNs face issues not present in residential settings, which have received more examination, and (ii) CGNs generally represent a second level of address translation—i.e., CGNs operate *in addition* to existing edge-based NAT—and therefore compound some of the issues that address translation raises.

While we know anecdotally that ISPs deploy CGN, we are not aware of quantitative studies of the prevalence and operation of CGNs in the wild. In this work, we take a first step toward developing an empirical understanding of these increasingly crucial pieces of Internet infrastructure. We make four high-level contributions:

Operator Perspectives on CGNs: We begin by presenting a survey of operators in § 2. We distributed a questionnaire on pertinent mailing lists, seeking to shed light on operators' motivations and experiences with CGN operation in the wild. We received illuminating input from 75 operators. Our survey reveals widespread adoption of CGN technology—with over half of the responding operators having deployed CGNs or planning to in the near future—despite the resulting operational difficulties.

Measurement Methodology: One of the key characteristics of CGNs is their *transparent* operation from the perspective of endpoints. While transparency has its benefits

(e.g., clients require no setup process to use a CGN), it complicates detection and measurement of CGNs. Multiple levels of address translation increase the difficulty further as each step overwrites any evidence a previous NAT left in the traffic. Therefore, the sender of a packet cannot tell if or how many times the source address will be translated on the path towards a destination, and the recipient cannot know the original source of the packet.

To address these difficulties we introduce two methods in § 4 for exploring CGNs. First, we observe that some nodes in the BitTorrent DHT mistake addresses internal to a CGN for external addresses and therefore propagate (“leak”) these to other nodes. Therefore, we are able to derive a broad understanding of the deployment of CGNs by probing the DHT. Our second set of methods relies on extensions to our Netalyzr measurement platform [24], which allow us to study the presence and detailed properties of CGNs based on locally available addressing information, repeated connectivity tests, as well as a new method that leverages the stateful nature of NATs and uses TTL-limited probes to force retention of state in some hops while allowing it to expire in others.

Studying Global CGN Presence: IPv4 address scarcity manifests differently for different networks in different parts of the world [36]. Our CGN detection methods give us a broad and unprecedented view into the global deployment of CGNs, which we present in § 5. Our vantage points cover more than 60% of the Internet’s “Eyeball ASes” that connect end users to the Internet. We find the CGN penetration rate to be 17–18% of all Eyeball ASes. Moreover, we find that CGN deployment is ubiquitous in cellular networks with more than 90% of all cellular ASes deploying CGNs. We also find a direct relationship between regions with higher perceived IPv4 address scarcity and CGN deployment.

Understanding CGN Behavior: CGNs present a massive resource distribution problem, whereby scarce public IPv4 addresses are multiplexed using a relatively small set of internal IPv4 addresses and a limited port space across thousands of end hosts. CGNs can be configured in a multitude of ways, with currently little known about CGN configurations, dimensioning, and behavior in the wild. Hence, in § 6 we make our final contribution: a deep dive into the properties of deployed CGNs. We analyze the internal address ranges used by CGNs, which reveals that some ISPs even face scarcity of internally used (“private”) address space. We also find CGN placement is diverse, ranging from 1–12 hops from the user. We find that the methods CGNs use to distribute available public IP addresses and port numbers to their subscribers vary dramatically. We then assess how CGNs restrict user connectivity and compare our insights about CGNs to the properties of commonly deployed CPE (customer premises equipment) NATs.

Finally, we note that while our study provides an unprecedented view into the use and properties of CGNs in the wild, we only partially illuminate the CGN landscape. Each of our measurement approaches has limitations that somewhat restrict their scope. For instance, since mobile devices rarely

use BitTorrent, our DHT crawl does not shed significant light on the use of CGNs within mobile ISPs. Our study constitutes an initial view into the deployment of CGNs with much future work to be done to better understand the impact of these critical components of the modern Internet.

2. AN OPERATOR’S PERSPECTIVE

To gain a better understanding of the real-world challenges that IPv4 address scarcity poses and how ISPs are coping, in late 2015 we circulated a survey on a dozen of network operator mailing lists and eventually collected responses from 75 ISPs located all over the world. These ISPs run the gamut in terms of size and type, including cellular and residential ones. While we do not claim the respondents form a statistically unbiased sample, we note that we received answers from operators in all regions of the world, spanning the whole spectrum of ISPs (cellular, residential) ranging from small rural ISPs in Africa up to Fortune 50 companies, connecting millions of subscribers to the Internet. Thus, we do believe that the approaches and concerns raised by these ISPs deserve our attention. Next, we summarize the survey responses.

IPv4 Address Space Scarcity: More than 40% of the responding ISPs indicate that they directly face IPv4 address scarcity issues. Some ISPs report a subscriber-to-IPv4 address ratio as high as 20:1. However, others point out that while their subscriber-to-address ratio is 1:1, internal subnetting and fragmentation make address space management cumbersome, especially when attempting to accommodate new customers. Another 10% of the respondents indicate that while they do not yet face scarcity, they believe it is looming in the near future. The ISPs not facing IPv4 address scarcity are mainly ones that received significant blocks of address space many years ago, as well as ISPs in the African region.¹ Interestingly, three ISPs also indicated that they face scarcity of *internal* IPv4 address space. These networks leverage CGN but also need internal address space for their internal management.

IPv4 Address Space Markets: Three of the responding ISPs report that they have bought IPv4 addresses, while another 15 ISPs indicate that they have considered procuring additional addresses. However, ISPs indicate concern regarding buying address space, including price of available address blocks (named by 60%), fear of obtaining “polluted” address blocks with a bad reputation from previous use (44%) and uncertainty regarding the ownership of blocks (42%).

CGN Deployment vs. IPv6 Deployment: Figure 1 shows the respondents’ approach to CGN and IPv6. Almost 40% of the ISPs indicate they deploy IPv4 CGNs, with another 12% considering CGN in the near-term. Typically, ISPs note incremental CGN deployments, either targeting new customers or shifting specific subsets of subscribers into CGN deployment. That is, *most CGN deployments are partial*.

¹Africa is the only region in which the IPv4 address pool is not yet depleted.

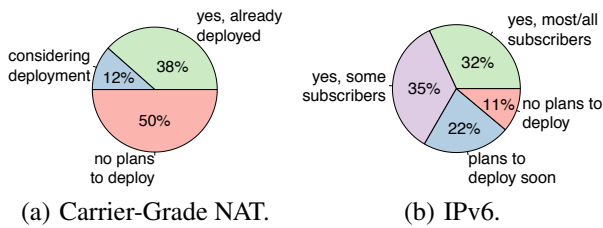


Figure 1: ISP survey: Status of Carrier-Grade NAT deployment and IPv6 deployment.

Next we find that 32% of the ISPs indicate IPv6 deployment to most or all of their subscribers, while another 35% have partial IPv6 deployments for some subscribers. The dominant transition mechanism noted is dual stack. Some ISPs also provide customers with an internal (CGN) IPv4 address and a publicly reachable IPv6 address. This arrangement will likely gain popularity in the near future as IPv4 connectivity will remain necessary until full IPv6 deployment.

CGN Concerns: Participating ISPs also had the option to inform us about possible concerns when operating CGNs. The responding ISPs raised several concerns regarding the setup and the operation of CGNs. A primary concern is that some applications (e.g., online gaming) do not work seamlessly with their CGN setups, causing subscriber complaints that remain difficult for the ISPs to resolve at the best of times. Additional concerns relate to traceability of users behind CGNs. Losing the ability to directly identify users can raise two kinds of problems. First, ISPs may be legally required to be able to map flows to subscribers. Second, IP addresses accrue reputations as they get used—e.g., as sources of spam—and therefore by sharing IP addresses among users the reputation is also shared and can cause problems for some users.

In addition, operators voiced concerns about a lack of well-developed best practices for configuring and dimensioning CGNs, rendering operating these devices cumbersome. In particular, operators need to resort to experimentation on aspects such as the distribution of external IP addresses and port ranges to customers, and whether to use distributed or centralized CGN infrastructure. Respondents named the port space as well as the amount of state CGNs need to maintain as primary challenges when configuring CGNs. Accordingly, ISPs report widely varying dimensioning of their CGNs in practice, ranging from static 1:1 NAT per customer—to prepare for the future—to limits of 512 sessions per customer due to heavy NATing.

3. BACKGROUND

As we sketch above, the lack of ready access to new IPv4 addresses is leading ISPs to alternate technologies to accommodate their addressing needs. One such approach is to leverage Carrier-Grade NATs (CGN). When an ISP uses CGN, it provides subscribers with internal IP addresses and then applies address translation to their traffic. CGNs often introduce multiple layers of address translation since sub-

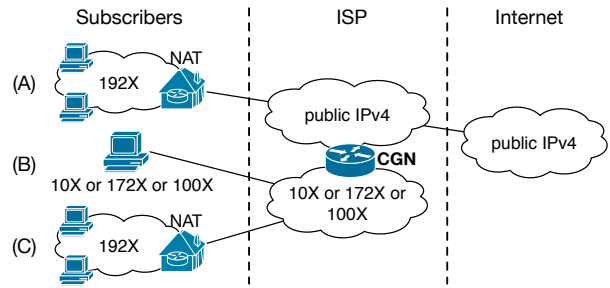


Figure 2: NAT scenarios. A resides behind a single in-home NAT, B behind a single carrier-grade NAT, and C behind both an in-home and carrier-grade NAT (NAT444).

Range	Shorthand	RFC	Comments
192.168.0.0/16	192X	1918	Commonly used in CPE
172.16.0.0/12	172X	1918	
10.0.0.0/8	10X	1918	
100.64.0.0/10	100X	6598	for CGN deployments

Table 1: Address space reserved for internal use.

scribers often run NAT devices on their own edge networks (e.g., as built into most CPE devices in users’ homes). We refer to the case of subscribers whose packets are translated once before they reach the public Internet as *NAT44* and to the case where packets are translated twice as *NAT444* [41,47].

Figure 2 illustrates various addressing structures in common use on the Internet. In each of the scenarios the ISP has a pool of public IPv4 addresses that are used differently by various subsets of its customers. The ISP gives each subscriber in group A a single public IP address. The subscriber in turn runs a subscriber-side NAT44 device to share this IP address among all the devices on the internal network. This is typical for many residential subscribers. Subscriber B receives an internal IP address from the ISP which a NAT translates into a public IP before packets reach the wide-area network. This case of a carrier-side NAT44 device is common within cellular networks. Finally, subscriber C’s network is identical to subscriber A’s in that a local NAT is used to facilitate connectivity for a multitude of internal devices. However, in this case, instead of providing a single external IP address the ISP provides the subscriber with a single internal IP address, which in turn it translates with a CGN before traffic reaches the wide-area network. This is a case of NAT444, or two layers of address translation. An ISP that runs a CGN does not necessarily NAT all of its subscribers. Many ISPs only NAT new subscribers and some even have various classes of subscribers and allow customers to choose their type of connectivity, which may come at different prices (some ISPs charge their customers for a public IP address, e.g., [31]).

On the basis of the terminology used in the IETF, we now define several NAT-related terms we use throughout the remainder of the paper.

Address Types: We distinguish IP addresses both in terms of their location relative to a NAT, as well as in terms of their numeric value. We refer to an address on the edge-facing, client-local side of a NAT as *internal* vs. *external* when nearer to the network core. An address is *reserved* if it resides in prefixes (as set forth e.g., by RFC 1918 [35]) that should not get announced to the global routing table, and *routable* otherwise. Table 1 lists those address ranges reserved for internal use by the IETF.²

NAT Mappings: NATs keep state that maps each internal IP address and port number tuple to an external IP address and port number tuple. Unless manually configured, NATs create mappings on-demand once a local host behind the NAT (i.e., with an internal IP address) sends a packet from its $IP_{int}:port_{int}$ endpoint to a remote $IP_{dst}:port_{dst}$. The NAT then records an $IP_{ext}:port_{ext}$ tuple, translates the packet and sends it to the destination host. When the external host replies to $IP_{ext}:port_{ext}$, the NAT finds the corresponding entry in its mapping table, translates the destination address to $IP_{int}:port_{int}$ and forwards the packet internally.

Mapping Types: NAT behavior differs in the reuse of existing mappings and in the filtering rules for the usage of established mappings. A *symmetric NAT* creates different mappings for subsequent packets sent from the same $IP_{int}:port_{int}$ endpoint to different $IP_{dst}:port_{dst}$ endpoints. This behavior significantly impedes NAT traversal and makes symmetric NATs the most restrictive type of NAT. Other types of NAT reuse existing mappings regardless of their $IP_{dst}:port_{dst}$. They differ in their filtering policy, here listed in decreasing order of restrictiveness: *port-address restricted NATs* only allow incoming packets from the very $IP_{dst}:port_{dst}$ that was initially contacted from the host inside the NAT, *address restricted NATs* require a matching IP_{dst} , but allow packets from varying port numbers, while *full cone NATs* allow incoming packets from any external host once a mapping is created. This makes full cone NATs the most permissive type of NAT [40].³

Mapping Timeouts: As with any stateful middlebox, NATs must manage their internal state and therefore track active flows. The NAT must release mappings that are no longer needed. NATs generally use both TCP state tracking and timeouts to prune unnecessary NAT mappings. Recommended minimum timeouts are 120 seconds for UDP [6] and 2 hours for TCP [21].

Port Allocation: NATs differ in their selection of an external $port_{ext}$ number for a new session. NATs implementing *port preservation* attempt to retain the original source port as the external port (i.e., $port_{int} = port_{ext}$), unless there is a collision and an alternate port must be chosen. Other NATs—especially large NATs—assign ranges of the external port space to each internal host and then assign external

ports on-demand from this pool in sequential or random order [6].

IP Pooling: Large NATs typically use multiple external IP addresses, called *NAT pooling*. Upon connecting, a subscriber typically gets allocated a public IP address out of the pool. NATs employing *paired pooling* always use the same IP_{ext} for a given IP_{int} . Otherwise, a NAT is said to use *arbitrary pooling*. In our methodology, the presence of NAT pools will play an important role when it comes to dissecting home NAT deployments from CGN deployments.

Hairpinning: Consider the communication between two hosts—*A* and *B*—behind the same NAT. When *A* sends a packet to *B* it will use *B*'s $IP_{ext}:port_{ext}$. When the NAT receives this packet it can detect that the destination of the packet is in fact itself and therefore direct the packet to *B*'s $IP_{int}:port_{int}$. This behavior is called *hairpinning* [6, 21]. If the NAT leaves the source $IP_{int}:port_{int}$ in place when forwarding the packet, then the hosts can discover their internal IP addresses when communicating behind the same NAT.

3.1 Related Work

IETF RFCs contain most of the available literature about CGNs. In particular, RFC 6888 specifies basic requirements for CGNs [34], whereas RFC 6544 [38] and RFC 5128 [43] describe two popular mechanisms for NAT traversal: ICE and UDP/TCP hole punching, respectively. As a result of NAT's added complexity, RFCs also describe how CGNs affect application-level functionality [17, 43].

Several academic studies have tried to identify NAT deployment in home networks using UPnP queries [16, 26] or IP ID header fields [7], by passively observing IP TTLs and HTTP User-Agent strings [28], and by applying NAT detection to NetFlow traces [25]. Müller *et al.* conducted an active, topology-based traversal of cascaded large-scale NATs [30]. One NAT test presented in our work is an augmented version of their methodology. Ford *et al.* studied the effectivity of different NAT punching techniques in NAT-ed networks [20]. The studies conducted by Wang *et al.* [45] performed a comprehensive active measurement campaign to understand middleboxes present in cellular networks. In contrast to Netalyzr their tool relied on rooted handsets to modify packets at the IP and TCP layers. Donley *et al.* [9] studied the impact of CGN deployment on Web browsing performance in one ISP. Ohara *et al.* [32] simulated how CGNs can impact on TCP connection establishment in mobile networks. Finally, Skoberne *et al.* presented a theoretical taxonomy of NAT deployments and discuss their possible impact on network performance [41]. Richter *et al.* [37] measured an increasing concentration of traffic on fewer IPv4 addresses during 2015, hinting at an increasing use of CGN deployment in the Internet.

Little is known about actual CGN hardware deployed in the wild and their consequences for the different stakeholders. We cannot readily identify how NAT vendors implement their equipment and how ISPs take advantage of them. To partly overcome this limitation, we rely on vendor manuals

²Technically some reserved addresses are in fact routable; we focus here on their intended use.

³This terminology allows arranging NATs according to their restrictiveness and improves readability, therefore we use it despite being discouraged by the IETF [6].

and network operator tutorials to obtain deeper insights into practical considerations of CGN deployment [2, 3, 12, 47].

4. DETECTING CGN AT SCALE

Our first set of methodologies aim to investigate the breadth of CGN deployment in the Internet. In general terms our CGN detection mechanisms leverage both internal and external observations of IP addresses associated with a given host to detect discrepancies and therefore presence of address translation. We use two techniques to obtain internal observations: implicit and explicit. Our implicit observations come via standard BitTorrent clients leaking internal address information, while our explicit observations come from users running active measurements on our behalf via our Netalyzr tool. We stress that we strive for conservativeness in our CGN detection methods. That is, we would rather provide a sound lower bound on CGN presence than using a more speculative approach that identifies more CGNs of questionable validity.

4.1 Detecting CGNs via BitTorrent

The BitTorrent Distributed Hash Table (DHT) [8] represents a distributed data structure that links hosts looking for specific content with hosts that have that content without using centralized infrastructure. The nodes that make up the DHT form a connected graph so that search queries for specific content are propagated to a node with the given information. Each node is identified by a 160 bit *nodeid* which is randomly chosen by the node itself (and is unique with high probability). To form the graph, DHT nodes both maintain a list of DHT peers and provide an interface for other nodes to query this list. Further, the nodes on this list must be periodically validated with *bt_ping* messages to ensure reachability. This in turn means that the contact information a node *A* has for node *B*—in the form of an IP address and port number—represents *B*'s location from *A*'s perspective. We observe that the nodes represent vantage points that we do not control but can none-the-less probe to learn about host-to-host connectivity. We find that this connectivity is sometimes represented by internal IP addresses. That is, the path between two hosts does not traverse the publicly routed Internet, but takes place completely within a private network (e.g., within an ISP). Additionally, these hosts are clearly also able to communicate outside of this private network and therefore are behind some form of NAT. We developed a crawler to collect connectivity information from the BitTorrent DHT and then leverage that data to form an understanding of CGN deployments.

Crawling the BitTorrent DHT: We developed a crawler that starts with DHT nodes learned from the BitTorrent bootstrap servers and issues a series of *find_nodes* requests to DHT nodes with a random query. The response to *find_nodes* is a list of up to eight “close” peers where closeness is calculated using the XOR distance between the query and each *nodeid* in the node’s list of peers [29]. We issue five queries, which provides connectivity information—*nodeid*, IP address and port number—for roughly 40 nodes. We then

	Peers	Unique IPs	ASes
Queried	21.5M	15.5M	18.8K
Learned	192.0M	62.1M	26.7K

Table 2: BitTorrent DHT data: *Queried*: Peers that were issued and replied to *find_nodes* requests. *Learned*: All peer information we gathered.

Range	Internal Peers		Leaking Peers		
	Total	Unique IPs	Total	Unique IPs	ASes
192X	565.9K	11.2K	186.8K	162.2K	4.1K
172X	336.6K	85.0K	52.9K	33.9K	1.0K
10X	1.3M	328.5K	283.9K	194.4K	2.2K
100X	1.5M	251.5K	192.0K	165.8K	723

Table 3: Peers reported via reserved IP addresses (left) and the corresponding peers that leaked them (right).

in turn query the newly learned peers in the same fashion. Our crawler also participates in the DHT and therefore accepts incoming requests from nodes that have learned about our crawler through the source information in our requests.

As we note above, in some instances peers reply to *find_nodes* with information about nodes that include reserved IP addresses (Table 1), indicating the probed peer can reach the reported peer without crossing the publicly routed Internet. We refer to this behavior as *internal address leakage*. When we learn an internal address for a given *nodeid* we refer to this node as an *internal peer*. When our crawler finds a node leaking internal peers we issue an additional ten *find_nodes* queries in the hopes of finding additional internal peers. We continue issuing *find_nodes* queries in batches of ten for as long as we continue to harvest internal peers.

Note that within BitTorrent the *nodeid* is the sole identity notion for a given peer. However, as peers can have multiple endpoints (internal, external), as well as multiple IP addresses/ports due to dynamic IP address assignment or due to BitTorrent clients modifying the local port number, we identify a unique peer by the full tuple of (*IPaddress:port, nodeid*). As a positive side effect, this also eliminates possible biases due to DHT poisoning [44], where peers announce themselves with a foreign *nodeid*.

The dataset we use in the remainder of this paper comes from a one-week crawl starting on March 3, 2016.⁴ Table 2 summarizes the dataset. We probed more than 21M peers across nearly 19K ASes. These probed DHT nodes in turn revealed contact information for 192M peers across more than 26K ASes. Of these 192M peers, 107.7M peers and 36.7M unique IP addresses responded to *bt_ping* probes. Table 3 shows an overview of the leaked contact information, where we break private peers down based to the internal address space range. Among the peers crawled, we find more than 700K peers leaking contact information for more than 3.7M internal peers (i.e., peers with IP addresses in a reserved range) across more than 5K ASes.

Identifying CGNs: Our dataset clearly shows the presence

⁴We have additional crawls from late 2015 and early 2016 that show consistent results to those we present in this paper.

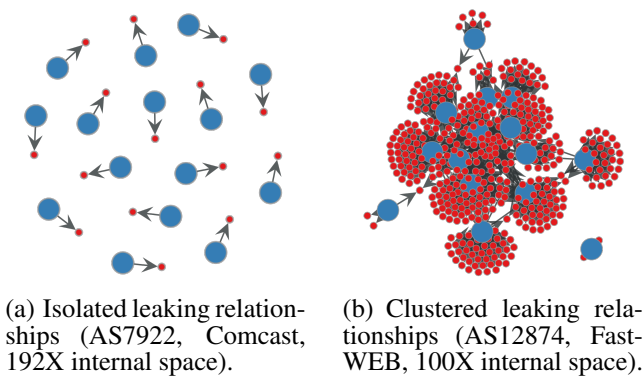


Figure 3: Peer leakage in non-CGN vs. CGN ASes. Large blue vertices are BitTorrent peers leaking peers with internal IP addresses (small red vertices).

of NATs via leaked internal peers. Next we seek to establish the degree to which these NATs are network-level CGNs as opposed to simple home NAT deployments. First, to detect any type of NAT using the BitTorrent dataset there must be multiple BitTorrent clients that directly communicate within some internal network. Next, to determine the presence of a large network-level CGN we require *NAT pooling* behavior (§ 3). In other words, within a single AS we require (i) multiple peers with different external IP addresses to leak internal peers and (ii) intersections in the internal peers leaked across multiple external IP addresses. Moreover, we require the internal peers within a cluster to reside within the same internal address range (e.g., 10/8).

To detect this behavior on a per-AS level we next form a graph for each AS where each peer is a vertex and each edge between a public peer A and an internal peer B indicates that A leaks contact information for B . Note, when constructing graphs we only consider internal peers which were leaked exclusively by peers residing in a single AS. This excludes leaking relationships caused by VPN tunnels. Figure 3 shows a small subset of the graphs for two ASes as an illustration. Figure 3(a) shows there is only isolated leaking within AS 7922 (Comcast). We find more than 1K peers leaking internal addresses within AS 7922. However, we also find that each leaked internal peer is leaked by exactly one external peer. In contrast, Figure 3(b) shows strong clusters within AS 12874 (FastWEB) consisting of multiple peers behind different external IP addresses that leak a large number of internal peers, which form large intersections.⁵ This shows that our clustering methodology is effective in separating home NAT deployments from network-level CGNs.

We next construct a graph for each AS in our dataset and determine the largest connected cluster for each AS. Figure 4

⁵We manually confirmed CGN presence in AS 12874 and also verified the discovery of internal peers (via *NAT Hairpinning*, § 3) and leakage of internal peers in this AS by running a regular BitTorrent client on a host behind CGN in this AS.

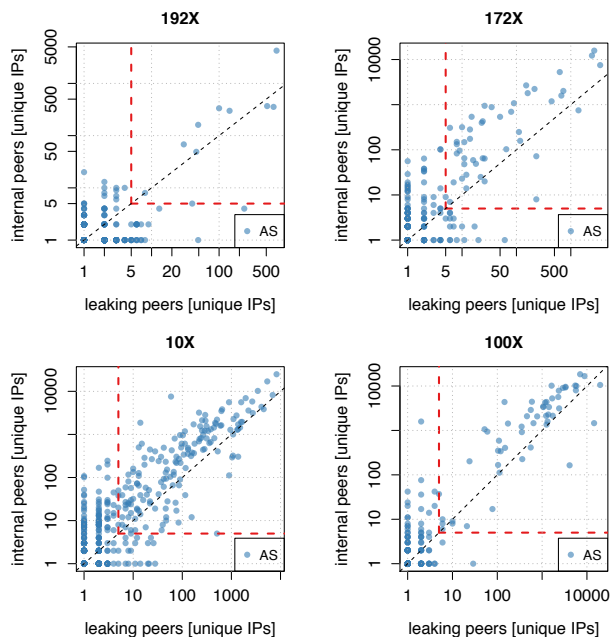


Figure 4: Size of the largest connected cluster of leaking and internal BitTorrent peers per AS. The x-axis shows the number of public IP addresses, the y-axis the number of internal IP addresses contained in the largest cluster.

shows our clustering results. Here, we plot a point for each AS and position it according to the size of the largest cluster we found in this AS (if any). In particular, the x -axis shows the number of unique public IP addresses contained in the largest cluster and the y -axis shows the number of unique internal IP addresses contained in the largest cluster. We only find a small number of ASes that contain large clusters in the 192X space (top left figure). We find ASes with large clusters to be more prevalent in the other, larger, internal address ranges. This supports our hypothesis that 192X address space is primarily used in small home NAT environments. While in principle a cluster with at least two different external IP addresses is indicative of *NAT pooling*, we only determine CGN presence for an AS when the largest connected cluster contains at least five public IP addresses and five private IP addresses. This is to address possible misclassifications arising from dynamic addressing, e.g., a home network with internal NAT deployment that changes its public IP address. We annotated Figure 4 with our detection boundary. While we show network-wide results in § 5, we note that this methodology shows CGN usage in roughly 10% of the probed ASes for which our crawler queries at least 200 peers.

DHT Data Calibration: Our BitTorrent-based CGN detection relies on three key properties of the DHT peers: (i) BitTorrent peers behind the same NAT can learn internal endpoints of other peers, (ii) peers export internal endpoints via the DHT, and (iii) peers only propagate contact information for peers that have been validated via direct interaction. We verified (i) and (ii) by running two popular and unmodified BitTorrent clients (uTorrent on Windows and Transmission

Address Space	Cellular	Non-cellular	
	IP_{dev} $N=8.6K$	IP_{dev} $N=567.5K$	IP_{cpe} $N=229.8K$
192X	0.2%	92.4%	8.9%
172X	2.5%	1.1%	0.8%
10X	58.7%	6.2%	4.8%
100X	17.3%	0.0%	1.9%
unrouted	12.5%	0.0%	0.0%
routed match	5.7%	0.0%	83.0%
routed mismatch	3.0%	0.3%	0.5%

Table 4: Address ranges seen for the device IP address and for the router’s external IP address.

on Linux) and measuring the control traffic they exchange as part of their regular operation. We confirmed that these peers learned their internal endpoints when located behind NATs that allow multicast communication as well as behind NATs that have *Hairpinning* enabled. Further, these peers forward packets with internal source IP addresses (see § 3). We also validated the latter within an ISP that deploys CGN. Therefore, we conclude that BitTorrent clients can—if the circumstances allow it—learn their internal endpoints and propagate that information via the DHT when requested.

Finally, we assume hosts follow the BitTorrent DHT specification [8] and only propagate reachability information for peers they learn after reachability has been directly validated by the host itself. Otherwise, hosts would propagate potentially dubious reachability information and likely we would detect CGN presence in practically any AS that hosts enough peers. To validate our assumption, we setup a common BitTorrent client (uTorrent on Windows) with a *nodeid* of ID_{us} and let it interact with the DHT. At the same time on a different host we crawled the DHT requesting ID_{us} . We queried 100K peers and were given contact information for ID_{us} by 1,387 peers. We found that only 18 of these DHT peers (1.3%) did *not* validate the reachability of ID_{us} before propagating the information. This shows that our assumption that DHT peers follow the specification and properly validate reachability before propagating contact information is sound.

4.2 Detecting CGNs via Netalyzr

To complement our observations from crawling the BitTorrent DHT, we leverage ICSI’s Netalyzr network troubleshooting service [24]. While the BitTorrent DHT provides a useful set of specific information from end hosts, Netalyzr allows us to define explicit tests we wish to run from end hosts. These tests interact with a suite of custom-built test and measurement servers and return the results to our data collection server. We developed a set of tests aimed at illuminating NAT behavior and deployed these in 2014. While Netalyzr provides the potential to gather much richer information than we find in the BitTorrent DHT, we are at the mercy of individual users to access Netalyzr and run the tests. Users run Netalyzr via one of three supported clients: a Java applet for Web browsers, a command-line client, or an Android client available in the Google Play store [1].

In the context of understanding CGNs, Netalyzr offers

two advantages over our BitTorrent crawl. First, since BitTorrent is not heavily used on mobile devices the Android version of Netalyzr extends our view into this important network type.⁶ Second, Netalyzr allows us to directly obtain the IP addresses used by the host, including (i) the local IP address of the device that executes Netalyzr, IP_{dev} , (ii) the external IP address of the CPE router device as learned via UPnP (where available), IP_{cpe} , and (iii) the public IP address as seen by our Netalyzr server, IP_{pub} . We categorize IP_{dev} and IP_{cpe} in four categories: (i) private address from one of the reserved blocks for this purpose, (ii) *unrouted* for addresses that are nominally public, but do not appear in the routing table, (iii) *routed match* for case where the address is routable, appears in the routing table and matches IP_{pub} (i.e., the non-NAT case) and (iv) *routed mismatch* for the case where the address is routable, appears in the routing table but does not match IP_{pub} .

Cellular Networks: Detecting the use of CGNs in cellular networks is straightforward in Netalyzr because there are no devices between the mobile device and the ISP and therefore the classification of the ISP-assigned IP_{dev} directly indicates the presence of a CGN.⁷ While straightforward, we require five observations from an AS before we include it in our study to ensure our conclusions are sound and are not the result of some unexpected behavior. The second column of Table 4 shows the breakdown of IP_{dev} for all our cellular Netalyzr sessions. We find 94% of the sessions—i.e., all cases except the *routed match* case—show a translated address. A first view of CGN deployment in cellular networks on a per-AS basis shows 63.8% exclusively assign internal IP addresses to mobile devices. Similarly, we find 6.0% of ASes exclusively assign public IP addresses to devices and show no signs of address translation. Meanwhile, 30.3% assign a mixture of internal IP address and public IP addresses to devices. We note that another 5.0% of ASes assign internal addresses from public blocks that are actually routed, but still perform address translation!

Non-Cellular Networks: We next shift to Netalyzr’s detection of CGNs in non-cellular networks. In these networks, IP_{dev} is often assigned by a device in the device’s network and therefore when CGNs are present there are multiple address translations happening in the path (NAT444). This in turn makes detection difficult. First, we winnow our analysis to ASes that have at least ten Netalyzr sessions in our dataset.⁸ The third column of Table 4 shows that IP_{dev} is

⁶Note, while mobile devices can join wifi networks we scope our measurements to those on cellular data networks.

⁷Exceptions could be caused by users manipulating their network access with VPN tunnels or by users who run their own cellular access point (e.g., femtocells). Netalyzr’s Android client collects enough data to allow us to prune such cases from our analysis.

⁸Note, we require more observations in the non-cellular case (ten) compared to the cellular case (five) because the situation is not as straightforward due to the presence of in-path network equipment in the edge network. This makes the breadth of behavior we observe larger and in turn we need more observations to draw sound conclusions.

	routed ASes (N=52K)		eyeball ASes, PBL (N=2.9K)		eyeball ASes, APNIC (N=3.1K)	
	covered	CGN-positive	covered	CGN-positive	covered	CGN-positive
BitTorrent	2,724 (5.2%)	254 (9.40%)	1,673 (57.7%)	180 (10.8%)	1,824 (59.6%)	204 (11.2%)
Netalyzr non-cellular	1,367 (2.6%)	195 (14.3%)	866 (29.8%)	151 (17.4%)	929 (30.4%)	174 (18.7%)
BitTorrent \cup Netalyzr	3,166 (6.0%)	421 (13.3%)	1,791 (61.7%)	306 (17.1%)	1,946 (63.6%)	350 (18.0%)
Netalyzr cellular	218 (0.4%)	205 (94.0%)	175 (6.0%)	162 (92.6%)	171 (5.6%)	161 (94.2%)

Table 5: Coverage and detection rates of our methods as fraction of all routed ASes, as well as Eyeball ASes, primarily connecting end users, as derived from PBL and APNIC.

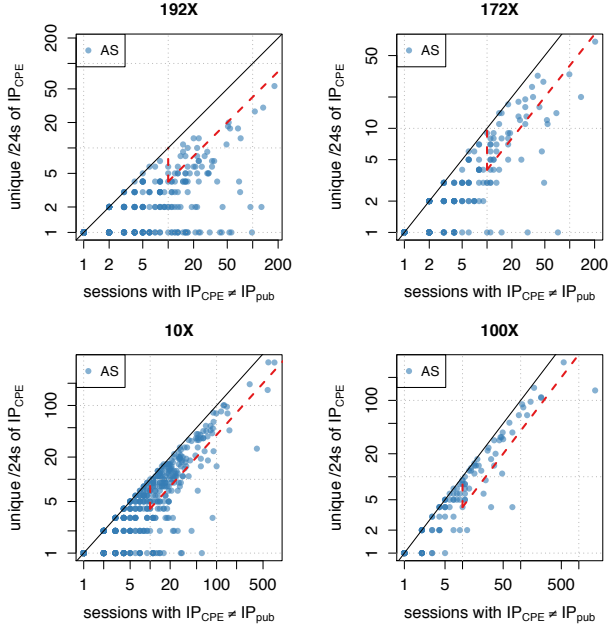


Figure 5: Netalyzr CGN candidate ASes: Sessions where IP_{cpe} does not match IP_{pub} (x-axis) vs. unique /24s of IP_{cpe} addresses (y-axis).

nearly always a private address, as expected. In addition to IP_{dev} , Netalyzr uses UPnP [10] to attempt to determine IP_{cpe} for the first hop CPE device. The fourth column of Table 4 shows the breakdown of IP_{cpe} for the 40% of cases where UPnP provides the address. In 83% of the cases, IP_{cpe} is a public IP address from the ISP, hence no CGN is present. The remaining 17% of the cases clearly point to multiple NATs. However, whether these are ISP-based CGNs or multiple small-scale NATs in the edge networks is not clear. Therefore, we add two steps to disambiguate the situation.

First, we observe that CPE routers often make assignments from the 192X block (Table 4, column 3), whereas the CGNs we find via BitTorrent and in the cellular environment more often make assignments outside the 192X block (Figure 4 and Table 4, column 2). Therefore, we use Netalyzr’s list of IP_{dev} assignments to determine the top ten /24 blocks from which CPE devices make assignments (covering 95% of assignments). We then conclude that any IP_{cpe} that falls within one of these blocks was likely assigned by another local CPE device and not a CGN. Applying this filter removes

more than half the ambiguous situations and leaves us with 7.9% of Netalyzr’s sessions that may be CGNs.

As a second step, we observe that due to their scale, CGNs necessarily must more broadly assign addresses than would be necessary in a small-scale edge network. Therefore, to conclude a CGN is present we require IP_{cpe} diversity within an AS. Specifically, an AS must have $N \geq 10$ Netalyzr sessions that may be behind a CGN. We expect that as the number of Netalyzr sessions increases, our observations of address space diversity will, as well. Therefore, those sessions must span at least $0.4 \times N$ internal /24 address blocks are deemed to indicate a CGN is deployed.⁹ Figure 5 shows a point for each AS in our dataset, with the x-axis showing the number of ambiguous multiple NATing situations we observe and the y-axis showing the number of /24 address blocks we observe within the AS. The dashed line represents our CGN detection cutoff point. Similar to our observations in our BitTorrent dataset, the 192X address space is sparsely used for CGNs, while more CGNs are present in the other reserved address blocks. Overall, our method detects CGN presence in almost 15% of the covered ASes.

Our CGN detection is no doubt imperfect. However, we note that our heuristics start with cases where our data *conclusively* indicates multiple address translators are present. Further, manual validation against our survey results, ISP’ websites and threads on operator mailing lists lends confidence to our conclusions. Finally, as we note above, our methods for labeling CGNs are conservative. For instance, there are points to the right of dashed line in Figure 5 that likely represent undetected CGNs. These points represent many Netalyzr sessions that show much address diversity—but not enough to meet our threshold. Our validations and conservative cutoffs leave us confident in the determinations we make, at the likely expense of not identifying all CGN deployments.

5. A NETWORK-WIDE VIEW OF CGN DEPLOYMENT

We now summarize our measurements of global CGN deployment based on the methodologies we develop in § 4. Table 5 reports our results in terms of ASes where we de-

⁹We note that we do not expect address diversity to infinitely scale with the number of observations. However, given our data this heuristic works well. Furthermore, adding additional complexity to the methodology without grounding in empirical observation is not useful.

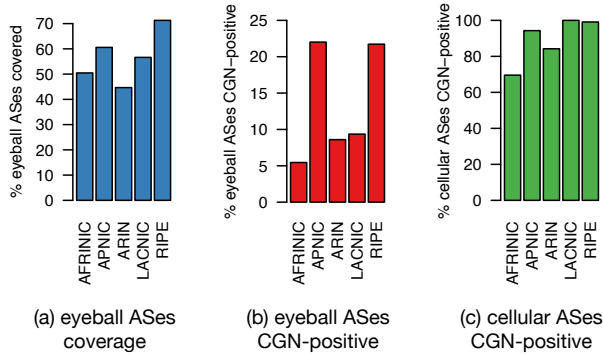


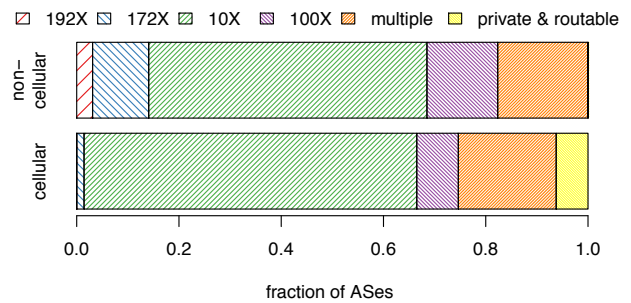
Figure 6: Eyeball AS coverage and CGNs per region.

tect at least partial CGN deployment. We report our results within the context of three different AS populations in the three big columns in the table. The second big column of the table considers the entire population of roughly 52K routed ASes. Meanwhile, the third and fourth columns represent the results in the context of so-called “eyeball” ASes (ASes that connect end users to the Internet). The third big column considers the population of ASes that the Spamhaus Policy Block List [42] identifies as including the equivalent of at least 2,048 addresses in “end user” blocks. The last big column considers the population of ASes to be those in the APNIC Labs AS Population list [5] that have at least 1,000 samples. Our datasets cover 6.0% of the ASes in the Internet, but over 60% of the eyeball ASes. Given that our methodologies rely on user-driven tools (Netalyzr and BitTorrent clients) it is unsurprising that we cover an order of magnitude more eyeball ASes.

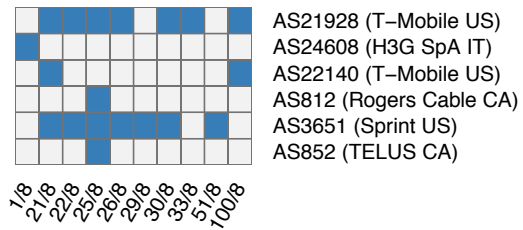
In terms of CGN deployments, we find that 13.3% of all non-cellular ASes use CGNs. However, the penetration jumps to 17–18% when considering only non-cellular eyeball ASes. In cellular networks the use is over 92% in all cases. These results show that CGNs are a reality for many Internet users. We also note that while we are able to cover roughly twice as many ASes with our BitTorrent dataset, the Netalyzr measurements find CGNs in higher proportions. This is expected and underscores important aspects of each methodology. While we are able to opportunistically leverage the information from the BitTorrent DHT, we are unable to direct or control the measurements. So, while BitTorrent has a large footprint the data is noisy. On the other hand, Netalyzr must coax people to explicitly run the tool and therefore the population is not large. However, once run we directly control the measurements and can gather more data directly (e.g., via probing UPnP). Finally, we note that the table shows that Netalyzr often does not add significantly to the coverage, but does add significantly to the CGN deployment results. Therefore, the BitTorrent detection should be viewed as a lower bound on CGN penetration.

Finally, we return to the impetus of NAT in the first place: address scarcity. Figure 6 shows our results by RIR.¹⁰ The

¹⁰The management of the address space is distributed over 5 Regional Internet Registries (RIRs). ARIN serves the North



(a) Internal address space usage per CGN deployment.



(b) ASes using routable space as internal space.

Figure 7: Internal address space in CGN deployments.

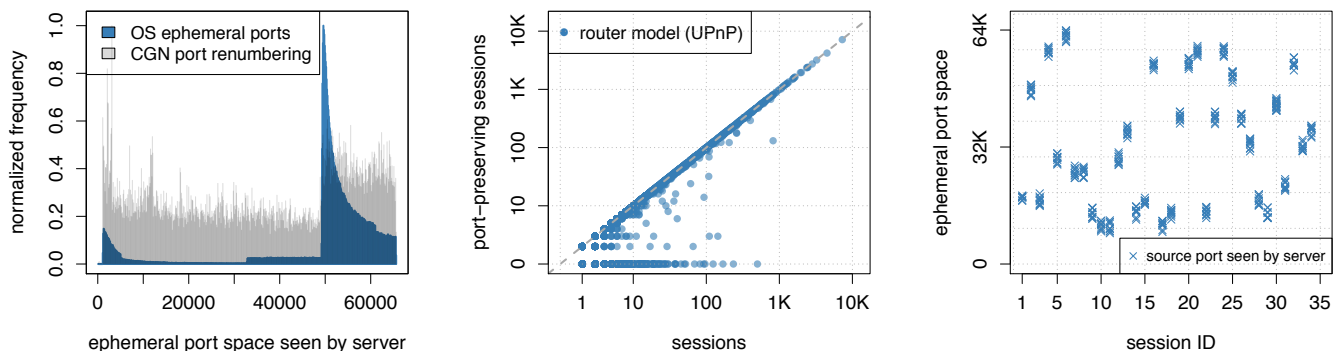
left-hand plot shows that the percentage of covered eyeball ASes within each region does not show a significant regional bias.¹¹ The middle plot in the figure shows the percentage of the eyeball ASes we find to deploy CGNs. Here we observe that APNIC and RIPE show more than twice the CGN penetration of the other regions. These are also the two regions that ran out of IPv4 addresses first. Meanwhile, we find the lowest CGN penetration in AFRINIC, which is the only region that has not yet exhausted its supply of IPv4 addresses. The last plot in the figure shows the CGN penetration in cellular networks by region. AFRINIC is again an outlier in this plot with “only” two-thirds of the ASes leveraging CGNs.

6. DRILLING INTO CGN PROPERTIES

Having a broad perspective of CGN deployment in today’s Internet in hand, we next drill into the properties of the detected CGNs. NATs can be configured in a multitude of ways and as our survey results indicate, configuring a NAT at carrier-scale presents a massive resource distribution problem, including (i) public IP addresses, (ii) private IP addresses and (iii) ephemeral port numbers. The CGN creates state in the form of NAT mappings with finite lifetimes (timeouts) to associate these resources depending on the NAT mapping type. A CGN’s configuration directly affects (i) the degree of resource sharing, i.e., how many subscribers can reside behind a given set of public IP addresses, as well as (ii) the number of simultaneous flows available to individual subscribers.

American region, LACNIC the Central- and South American region, RIPE the European region, AFRINIC the African region, and APNIC the Asian region. For more information, we refer to [36].

¹¹We use the PBL eyeball AS list for this plot.



(a) Ephemeral port space seen by our server from non-CGN vs. CGN connections without port preservation.

(b) Port preservation behavior of CPE equipment. 92% of UPnP sessions are from devices behind port-preserving CPEs.

(c) Observed ports per session, chunk-based random allocation strategy (4K ports per subscriber, AS12978).

Figure 8: Port Allocation Properties.

In this section, we study the configuration of our identified CGNs. In particular, we study (i) which address ranges ISPs use internally, (ii) how CGNs assign IP addresses and ports to their subscribers, (iii) topological properties of CGNs (i.e., the location of the NAT), and (iv) the kind of NAT mappings deployed CGNs commonly employ. Where appropriate, we contrast findings for CGNs with our findings for commonly deployed CPE devices.

6.1 Internal Address Space Usage

Our two probing methods enable us to evaluate properties of the address space behind detected CGNs. Figure 7(a) shows per AS the internal address space ranges used within non-cellular as well as cellular CGNs. Overall, we observe that naturally the largest private range (10X) is the most commonly used space for CGNs, followed by the 100X block newly allocated specifically for CGN deployments [46]. We also observe CGNs deploying the smaller 172X and 192X address spaces. Interestingly, roughly 20% of the ASes use multiple ranges of reserved address space in their CGN deployment. We speculate that the size of individual blocks does not suffice or, more likely, that such private address space is already in use in other parts of the organization. Some cellular ISPs in fact use routable address space for their internal CGN deployments. In Figure 7(b) we show which routable address blocks make up the most prominent cases we detected. While most of the routable address space used is not routed in practice (such as the 25.0.0/8 block, allocated to the UK Ministry of Defense), some ISPs use address space within their internal deployment (e.g., 1.0.0/8) that is publicly routed by other ASes. We contacted a representative of one of these major ISPs who confirmed that their internal deployment of routable address space results from scarcity of internal address space. Thus, some ISPs evidently *experience a shortage of internal address space* and adopt drastic measures at the expense of potential security and connectivity problems once public and internal addressing collides. Moreover, this address range proliferation renders troubleshooting CGNs even more cumbersome.

6.2 Port & IP Address Allocation

Next, we study how CGNs allocate ports and IP addresses to their subscribers. We start with the former. NAT port allocation may adopt the following strategies [6]: (i) *port preservation*, where the NAT attempts to maintain the local port of the flows; (ii) *sequential use*, where the NAT allocates ports in a sequential order; (iii) *random use*, where the NAT allocates ports without a clearly identifiable pattern.¹²

Measuring port translation: During one execution of Netalyzr (a “session”), its client opens 10 sequential TCP flows to an echo server listening on a high port number unlikely to be proxied. These TCP flows enable us to reason about the port allocation strategy implemented by the CGN, by comparing the local ephemeral port number, as chosen by the device, with the source port as seen by our server. Figure 8(a) shows the distribution of source port numbers as observed by our server. We show two histograms, one for port-preserved sessions and one for sessions exhibiting port translation. While operating systems employ ephemeral port ranges [13], CGNs translating port numbers utilize the entire port space. This observation could prove useful for server-side attempts (e.g., by content providers) to identify whether a client-side IP address belongs to a CGN.

Port translation of CPE routers: In non-cellular networks, where users’ packets typically get NATed by a CPE, our measurements might be affected by CPEs employing port translation. To assess the impact of CPEs on port translation, Figure 8(b) shows for each CPE model (inferred using UPnP) the number of *non-CGN* sessions where our server saw the same ports as chosen by the device. We observe that in more than 92% of non-CGN sessions the CPE did *not*

¹²We allow some leeway in determining port behavior. For example, we identify port preservation if at least 20% of ports remain preserved, and we declare *sequential use* if every two subsequent connections exhibit a numerical port difference smaller than 50. This accounts for situations in which NATs can not allocate the original or subsequent sequential port because of already existing mappings.

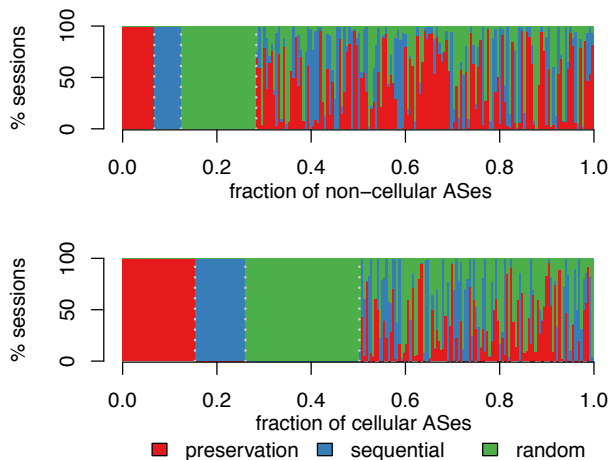


Figure 9: Distribution of observed port allocation strategies per CGN AS.

alter the source port numbers. Hence, while some CPE do translate ports, their effect on our analysis remains small.

Network-wide port allocation strategies: Figure 9 shows the distribution of port allocation strategies for each CGN-positive AS. We sort ASes with a “pure” allocation strategy (left part of the plot) and move ASes with mixed allocation strategies to the right. We observe a uniform port allocation strategy for about a third of the non-cellular ASes and for about 50% of the cellular ones. For the rest, *CGN behavior is heterogeneous*. We can attribute this to distributed CGN deployments, where users of the same ISP reside behind multiple CGNs, and the fact that NAT devices do not necessarily behave consistently, changing their behavior under load and over time [27]. Table 6 summarizes the dominating port allocation strategy per AS.

Chunk-based port allocation: In addition to the classification in our three categories of port allocation strategies, we also identify CGNs with random chunk-based allocation, where each subscriber receives a fixed port block [12]. Given sufficient data, we can infer the size of such port “chunks”. Figure 8(c) shows an example of chunk-based port management: AS12978 allocates 4K ports per subscriber. For each recorded session in this AS, our server observes source ports translated in no particular order (neither preserved nor sequential) and that all ports of a given session fall within a well-defined range.

In order to detect chunk-based allocation for all ASes, we require (i) at least 20 sessions with random port translation and (ii) all sessions exhibiting port numbers within a range smaller than 16K ports. Using this approach we identified 17 ASes using chunk-based allocations, shown in Table 6. While a minority of the identified CGNs, it allows us to reason directly about the dimensioning of the CGN e.g. in terms of the number of subscribers sharing a given IP address. We find 6 ASes in which subscribers only receive a port chunk smaller than 1K; for 3 of them, the chunk size falls to 512 ports—a scarily small number given that loading

Port allocation strategy	Non-cellular	Cellular
Port-preservation	41.2%	27.9%
Sequential	22.2%	26.0%
Random	35.6%	44.7%
Random (with chunk allocation)	9 (4.6%)	8 (3.7%)
Chunk size (CS)		
$CS \leq 1K$	4	2
$1K < CS \leq 4K$	2	3
$4K < CS \leq 16K$	3	3

Table 6: Port allocation strategies observed for CGN ASes. For ASes implementing chunk-based random port allocation we estimate the per-subscriber chunk size.

a single Web page can result in many dozens of TCP connections to fetch its various objects [11], resulting in a sizeable overall number of concurrent connections in residential networks [4]. The size of the port chunks then also directly translates into the maximum number of users per public IP address: we find 64 subscribers per IP address in the case of a 1K port chunk.

NAT pooling behavior: For the majority of the CGN-positive ASes, we observe *paired pooling behavior*, i.e., subsequent TCP sessions are bound to the same public IP address (recall § 3), with varying port allocation patterns. However, we find that 21% of the CGNs also employ *arbitrary pooling behavior*, i.e., Netalyzr reported multiple global IP addresses during the duration of the test for more than 60% of the sessions. This list includes major ISPs spanning all geographic regions. IETF guidelines [6] discourage this behavior due to its complicating effect on applications (particularly SIP and RTP), which use multiple ports on the same end host but do not negotiate IP addresses individually [6, 23].

6.3 CGN-specific measurements

To extract additional details about the CGNs under study, we extended our Netalyzr test suite with two tests. In this section we describe the tests’ operation; the corresponding findings follow subsequently.

TTL-driven NAT enumeration: An extension of previous work [30], this method identifies the precise on-path location and the mapping timeouts of cascaded NATs. To do so, it leverages the stateful nature of NATs and their need to remove the state of idle UDP flows from the translation table (recall § 3). During the test we repeatedly perform a reachability experiment that selectively detects stateful middleboxes in a chosen subset of the path, as depicted in Figure 10. As annotated, the test consists of three stages: (a) the client creates a UDP flow to our server, (b) both endpoints transmit TTL-limited “keepalive” probes for an idling period t_{idle} in order to keep the flow’s state alive up to but not at the hop under test, (c) the server checks whether it can still reach the client. If not, we conclude that the hop under test is a NAT that has removed the flow’s state. Our test enumerates the path between client and server by systematically performing iterations of this reachability experiment using different parameters for the client-side initial TTL t_{tlc} , server-side initial TTL t_{tls} , and elapsed idle time t_{idle} .

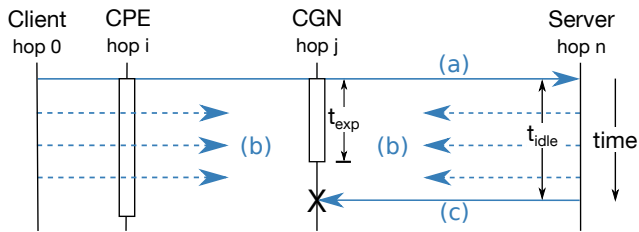


Figure 10: Reachability experiment scenario, consisting of initialization packet (a), keepalive packets (b), and probe packet (c). This example uses the parameters $i \leq ttl_c < j$ and $ttl_s < n - j$. If $t_{exp} < t_{idle}$ the mapping in the CGN (hop j) expires before the server replies with a probe packet.

	CGN detected	No CGN detected
IP address mismatch	67.6%	30.9%
IP address match	0.5%	0.9%

Table 7: Detection rate of TTL-driven NAT enumeration.

We acknowledge three limitations of this approach. First, as a crowd-sourced test relying on user involvement, we need to limit the idle period of the test. We test idle times up to 200 seconds, the maximum possible value without prolonging the overall runtime of a Netalyzr test session. Hence, NATs with a mapping timeout larger than 200 seconds go unnoticed, leading to an underestimation of the actual number of NATs: in 30.9% of the tests (see Table 7) we do not find an expired mapping, while a mismatch between the client’s local and server-perceived IP addresses (§ 4.2) evidently indicates NAT deployment. In the following, we only consider cases in which we could successfully observe an expired mapping. Second, based on the results of the reachability experiment we cannot distinguish between NATs and other stateful middleboxes such as stateful firewalls. However, we find stateful middleboxes without address translation in only less than 0.5% of our tests (see Table 7). We exclude these cases in the following analysis. Third, for a reliable expiration of the keepalive packets, the technique requires stable path lengths. Due to the large number of reachability experiments per test session (~ 60), we can detect and filter results with unstable paths.

STUN test: To study the mapping types of CGNs, we implemented a STUN [39] test in our Netalyzr test suite in October 2015. STUN determines the mapping type implemented by on-path NATs. STUN sends probe packets to a public STUN server (which answers certain probe packets from a different port and/or IP address) and waits for the respective replies.¹³

From the TTL-driven NAT enumeration test (deployed in September 2014) we have collected more than 38K sessions, whereas the STUN test (deployed in October 2015) produced 23K sessions. To be able to contrast sessions from within CGN-positive networks against CGN-negative ones,

¹³For more details on the operation of STUN, we refer to [39].

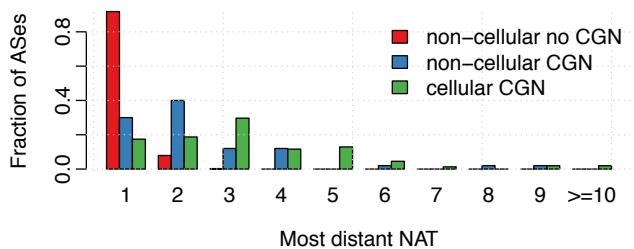


Figure 11: Maximum NAT distance from the subscriber.

we augment the results from both tests with the results from our CGN detection tests (§ 4). We further apply filtering rules to the results of both tests to ensure that we have collected at least three sessions from a particular network (combination of AS number and CGN classification type, e.g. “cellular CGN”). After applying the filters, this leaves us with 18K sessions from the NAT enumeration test running via both non-cellular (70%) and cellular networks (30%). The results cover 608 ASes, whereof 43% (259 ASes) deploy CGN. For the STUN test we count 20K sessions from non-cellular (87%) and cellular networks (13%). The STUN results span 720 ASes including 170 CGN-positive ASes (24%).

6.4 Topological Properties of CGNs

Figure 11 shows the distribution of the number of hops between the client and the most distant NAT detected, grouped per AS and its respective CGN deployment status. We detected NATs as far away as 18 hops from the client. As expected, most of the NATs in CGN-negative ASes (92%) sit just one hop away from the client, i.e. they are typically located right on the CPE router. Compared to that, most CGNs are located two to five hops away from the client (64% of non-cellular and 73% of cellular ASes). In non-cellular ASes the CGN distance mostly ranges from two hops up to six hops. In the case of cellular ASes, however, the CGN distance ranges from one hop to two hops and up to 12 hops away from the client. In fact, we find that for 10% of the cellular ASes, the CGN is located six or more hops away from the client. A large number of hops between client and CGN hints at a centralized CGN infrastructure with large aggregation points, which has the potential of affecting the accuracy of IP geolocation databases when locating the external IP address of clients behind CGN.

6.5 Flow-Mapping Properties of CGNs

The type of NAT mapping (recall § 3) as well as its state-keeping timeout directly affect the reachability of a host located behind a NAT, and thus has a profound effect on applications that rely on peer-to-peer connectivity [15, 20] or long-lived sparse flows [45].

Mapping Timeouts: Figure 12 shows the UDP mapping timeouts for the detected CGNs, both in the cellular, as well as in the non-cellular case. Here, we aggregate our CGN-positive sessions on a per-AS level. An AS is represented by its most frequent timeout value (mode). We also report

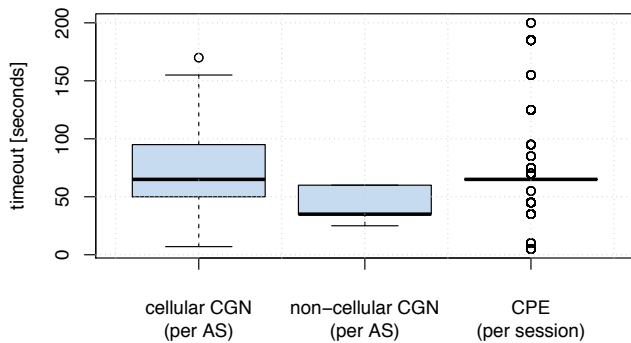
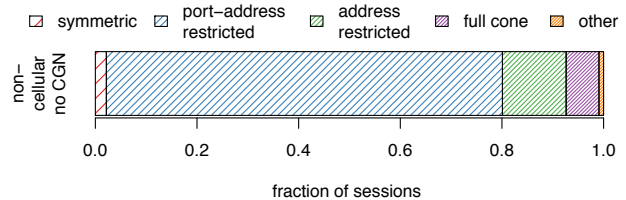


Figure 12: UDP mapping timeouts of CPEs and CGNs.

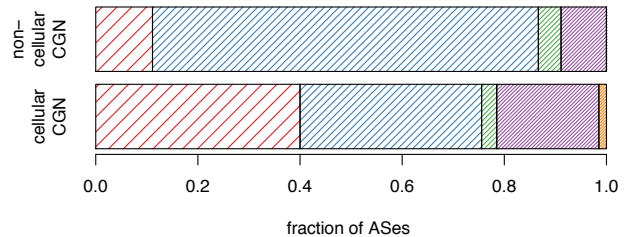
timeout values that we detected for CPE devices (shown in the right boxplot), where we show a boxplot of all recorded sessions. In NAT444 scenarios (non-cellular CGN) we need to make sure to report the timeout of CGNs rather than the CPE NATs. Therefore, to reason about CGN mapping timeouts, we only consider sessions that were detected as CGN (§ 4) and where our TTL-driven NAT enumeration detected the NAT at a distance of three or more hops away from the client. We observe that 74% of detected NATs expire idle UDP state after 1 minute or less, but we find values ranging from 10s to 200s.¹⁴ CGNs in cellular networks exhibit a larger median mapping timeout (65s) compared to non-cellular networks (35s). For CPE NATs we predominantly measured a timeout of 65s. We find higher variability and a lower median of timeout values for non-cellular CGNs when compared to CPE NATs. Low CGN timeout values might in turn negatively affect the longevity of sparse UDP flows that are also exposed to CPE NATs. While we find lower timeout values for CGNs compared to CPEs, we acknowledge that this property does not necessarily hold true for CGNs in general, as our test can not detect timeout values larger than 200 seconds.

Mapping types: Figure 13 shows our STUN results. We order the observed mapping types from most restrictive (*symmetric NAT*) to most permissive (*full cone NAT*). In Figure 13(a) we show the NAT mapping type as observed for CPE routers, while the bars in Figure 13(b) indicate the most permissive type of NAT mapping for our CGN-positive ASes. Recall that when multiple NAT devices reside on the path, STUN reports the most restrictive behavior of them, which also determines eventual NAT traversal. Hence, we argue that the most permissive STUN type provides a good approximation for the CGN behavior, because there cannot be a STUN result less restrictive than the CGN. We observe that, while exhibiting some diversity, less than 2% of the tests showed CPE NATs with very restrictive symmetric NATs. In contrast to CPE NATs, we observe 11% of non-cellular CGN ASes whose most permissive mapping type is symmetric. Among these networks we find many popular large European ISPs. For cellular networks we observe a bimodal

¹⁴Note that our timeout detection mechanism uses a 10 second probing interval. Hence, reported values can differ up to 10 seconds from the actual NAT timeout.



(a) Distribution of observed STUN types in CPE NATs.



(b) Most permissive STUN type per AS (only CGN sessions).

Figure 13: STUN results per AS.

outcome, with a large fraction of both restrictive (40% symmetric) and permissive (20% full cone) NAT types. We see large operators on both ends of the spectrum, with major cellular networks in the US deploying CGNs with symmetric mapping types.

Thus, we often measure stricter NAT mapping policies for CGN-positive sessions when compared to common home CPE devices. We conclude that a large fraction of ISPs deploy CGNs that use symmetric flow mappings, which limits the customers’ ability to establish direct connections. For this reason, the IETF lists an endpoint-independent mapping (which symmetric NATs violate) as their first requirement for CGNs [6, 21].

7. IMPLICATIONS

Our analysis shows that ISPs widely deploy CGN. We find that more than 17% of eyeball ASes and more than 90% of cellular ASes rely on CGNs (§ 5), with particularly high deployment rates in Asia and Europe—regions in which IPv4 address scarcity cropped up first, as the respective registries ran out of readily available IPv4 addresses in 2011 and 2012. Thus, adopting CGN presents a viable alternative to buying IPv4 address space from brokers. CGNs actively extend the lifetime of IPv4 and hence also fuel the demand of the growing market for IPv4 address space [36], which in turn affects market prices and possibly hampers the adoption IPv6.

CGNs directly affect “how much Internet” a subscriber receives, by (i) limiting available ephemeral port space, (ii) restricting the directionality of connections, and (iii) limiting connection lifetimes due to finite state-keeping budgets. Studying our identified CGN deployments, we find a wide spectrum of configurations and degrees of address sharing (§ 6). On the limiting end of the spectrum, we find ISPs allocating as little as 512 ephemeral ports per subscriber (§ 6.2), multiplexing up to 128 subscribers per public IP address. Comparing NAT flow mapping types and timeouts of

CGNs to commonly deployed CPE hardware, we find that in many instances CGNs use more restrictive flow mapping types when compared to their home counterparts (§ 6.5). This rules out peer-to-peer connectivity, complicating modern protocols such as WebRTC [22] that now need to rely on rendezvous servers.

We argue that the lack of *guidelines* and *regulations* for CGN deployment compounds the situation. While the IETF publishes best practices for general NAT behavior [6, 21, 33] as well as basic requirements for CGN deployments [34] (which, incidentally, many of our identified CGNs violate), dimensioning NATs at carrier-scale in a way that minimizes collateral damage remains a black art. Our finding that some large ISPs find the need to employ publicly routable (indeed, sometimes routed) address space for internal CGN deployment (§ 6.1) underlines the graveness of the situation. While it remains out of scope for this work to precisely measure the effect of CGNs on end-users' applications, we believe that our observations can serve as input for establishing such guidelines. Our findings should also interest regulators, who in some countries already impose acceptable service requirements on Internet performance (e.g., the FCC's measurements of advertised vs. achieved throughput [19]). We argue that the presence and service levels of CGNs should be readily identifiable in ISPs' offerings. Unfortunately, we find that most ISPs do not cover CGN deployment in their terms of service. Lastly, our findings document further erosion of the meaningfulness of IP address reputation, address-based blacklisting, IP-to-user attribution, and geolocating end-users (§ 6.2, § 6.4), which become all but infeasible in the presence of CGN.

8. CONCLUSION

This work presents a solid first step towards a better understanding of the prevalence and characteristics of CGN deployment in today's Internet. Our methods, based on harvesting the BitTorrent DHT and extensions to our Netalyzr active measurement framework, prove effective in uncovering CGN deployments: we detect more than 500 instances in ISPs around the world. When assessing the properties of these CGNs we find striking variability in the dimensioning, configuration, placement, and effect on the subscriber's connectivity. We hope this study will stimulate a much-needed discussion about best practices, guidelines, and regulation of CGN deployment. In future work, we plan to further improve our detection mechanisms and to study the impact of CGN on application performance.

Acknowledgments

We thank the network operators who participated in our survey for their insightful feedback and comments. We thank Daniele Iamartino for support with validating the BitTorrent data, and Martin Ott for his support with implementing the BitTorrent crawler. We also thank the anonymous reviewers for their helpful feedback. This work was partially supported by the US National Science Foundation under grants CNS-1111672 and CNS-1213157, the Leibniz Prize project funds

of DFG/German Research Foundation (FKZ FE 570/4-1), and the BMBF AutoMon project (16KIS0411). Any opinions expressed are those of the authors and not necessarily of the sponsors.

9. REFERENCES

- [1] Netalyzr for Android. Google Play. <https://play.google.com/store/apps/details?id=edu.berkeley.icsi.netalyzr.android>.
- [2] A. Tabdili. Carrier Grade NAT: Requirements and Challenges in the Real World. <http://www.menog.org/presentations/menog-10/Amir%20Tabdili%20-%20Carrier%20Grade%20NAT.pdf>.
- [3] A10 Networks. Carrier Grade NAT (CGN) Deployment Guide. https://www.a10networks.com/sites/default/files/resource-files/A10-DG-Carrier_Grade_NAT_%28CGN%29_Large_Scale_NAT_%28LSN%29.pdf.
- [4] S. Alcock, R. Nelson, and D. Miles. Investigating the Impact of Service Provider NAT on Residential Broadband Users. *TR, University of Waikato*, 2010.
- [5] APNIC Labs. Customers per AS Measurements. Description: <https://labs.apnic.net/?p=526> Dataset: <http://stats.labs.apnic.net/aspop>.
- [6] F. Audet and C. Jennings. Network Address Translation (NAT) Behavioral Requirements for Unicast UDP. RFC 4787 (Best Current Practice), January 2007. Updated by RFCs 6888, 7857.
- [7] S. M. Bellovin. A Technique for Counting NATted Hosts. In *IMW*, 2002.
- [8] BitTorrent.org. DHT Protocol (BEP-05). http://www.bittorrent.org/beps/bep_0005.html.
- [9] E. Bocchi, A. S. Khatouni, S. Traverso, A. Finamore, V. D. Gennaro, M. Mellia, M. Munafò, and D. Rossi. Impact of Carrier-Grade NAT on Web Browsing. In *IWCMC*, 2015.
- [10] M. Boucadair, R. Penno, and D. Wing. Universal Plug and Play (UPnP) Internet Gateway Device - Port Control Protocol Interworking Function (IGD-PCP IWF). RFC 6970 (Proposed Standard), July 2013.
- [11] M. Butkiewicz, H. V. Madhyastha, and V. Sekar. Understanding Website Complexity: Measurements, Metrics, and Implications. In *IMC*, 2011.
- [12] Cisco. NAT Administration Guide, StarOS Release 17. http://www.cisco.com/c/dam/en/us/td/docs/wireless/asr_5000/17-0/PDF/17-NAT-Admin.pdf.
- [13] Cymru. Ephemeral Source Port Selection Strategies. <https://www.cymru.com/jtk/misc/ephemeralports.html>.
- [14] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey. Measuring IPv6 Adoption. In *ACM SIGCOMM*, 2014.
- [15] L. D'Acunto, J.A. Pouwelse, and H.J. Sips. A measurement of NAT & Firewall Characteristics in Peer to Peer Systems. In *ASCI*, 2009.

- [16] L. DiCioccio, R. Teixeira, M. May, and C. Kreibich. Probe and Pray: Using UPnP for Home Network Measurements. In *PAM*, 2012.
- [17] C. Donley, L. Howard, V. Kuarsingh, J. Berg, and J. Doshi. Assessing the Impact of Carrier-Grade NAT on Network Applications. RFC 7021 (Informational), September 2013.
- [18] K. Egevang and P. Francis. The IP Network Address Translator (NAT). RFC 1631 (Informational), May 1994. Obsoleted by RFC 3022.
- [19] FCC. Measuring Broadband America. <https://www.measuringbroadbandamerica.com/>.
- [20] B. Ford, P. Srisuresh, and D. Kegel. Peer-to-Peer Communication Across Network Address Translators. In *USENIX ATC*, 2005.
- [21] S. Guha, K. Biswas, B. Ford, S. Sivakumar, and P. Srisuresh. NAT Behavioral Requirements for TCP. RFC 5382 (Best Current Practice), October 2008. Updated by RFC 7857.
- [22] C. Holmberg, S. Hakansson, and G. Eriksson. Web Real-Time Communication Use Cases and Requirements. RFC 7478 (Informational), March 2015.
- [23] Alan B. Johnston. *SIP: Understanding the Session Initiation Protocol*. Artech House, Inc., Norwood, MA, USA, 3rd edition, 2009.
- [24] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Notalyzr: Illuminating The Edge Network. In *IMC*, 2010.
- [25] V. Krmicek, J. Vykopal, and R. Krejci. NetFlow Based System for NAT Detection. In *ACM CoNEXT*, 2009.
- [26] A. Lutu, M. Bagnulo, A. Dhamdhare, and k. claffy. NAT Revelio: Detecting NAT444 in the ISP. In *PAM*, 2016.
- [27] D. MacDonald and B. Lowekamp. NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN). RFC 5780 (Experimental), May 2010.
- [28] G. Maier, F. Schneider, and A. Feldmann. NAT Usage in Residential Broadband Networks. In *PAM*, 2011.
- [29] P. Maymounkov and D. Mazieres. Kademia: A Peer-to-Peer Information System Based on the XOR Metric. In *Peer-to-Peer Systems*. Springer, 2002.
- [30] A. Müller, F. Wohlfart, and G. Carle. Analysis and Topology-based Traversal of Cascaded Large Scale NATs. In *ACM HotMiddlebox*, 2013.
- [31] NOW (New Zealand ISP). What if I need a public IP Address? <https://support.nownz.co.nz/support/solutions/articles/5000504832-what-if-i-need-a-public-ip-address->.
- [32] Y. Ohara, K. Nishizuka, K. Chinen, K. Akashi, M. Kohrin, E. Muramoto, and S. Miyakawa. On the Impact of Mobile Network Delays on Connection Establishment Performance of a Carrier Grade NAT Device. In *ACM AINTEC*, 2014.
- [33] R. Penno, S. Perreault, M. Boucadair, S. Sivakumar, and K. Naito. Updates to Network Address Translation (NAT) Behavioral Requirements. RFC 7857 (Best Current Practice), April 2016.
- [34] S. Perreault, I. Yamagata, S. Miyakawa, A. Nakagawa, and H. Ashida. Common Requirements for Carrier-Grade NATs (CGNs). RFC 6888 (Best Current Practice), April 2013.
- [35] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918 (Best Current Practice), February 1996. Updated by RFC 6761.
- [36] P. Richter, M. Allman, R. Bush, and V. Paxson. A Primer on IPv4 Scarcity. *ACM CCR*, 45(2), 2015.
- [37] P. Richter, G. Smaragdakis, D. Plonka, and A. Berger. Beyond Counting: New Perspectives on the Active IPv4 Address Space. In *ACM IMC*, 2016.
- [38] J. Rosenberg, A. Keranen, B. B. Lowekamp, and A. B. Roach. TCP Candidates with Interactive Connectivity Establishment (ICE). RFC 6544 (Proposed Standard), March 2012.
- [39] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing. Session Traversal Utilities for NAT (STUN). RFC 5389 (Proposed Standard), October 2008. Updated by RFC 7350.
- [40] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs). RFC 3489 (Proposed Standard), March 2003. Obsoleted by RFC 5389.
- [41] N. Skoberne, O. Maennel, I. Phillips, R. Bush, J. Zorz, and M. Ciglaric. IPv4 Address Sharing Mechanism Classification and Tradeoff Analysis. *IEEE/ACM ToN*, 2014.
- [42] Spamhaus. The Policy Block List. <https://www.spamhaus.org/pbl/>.
- [43] P. Srisuresh, B. Ford, and D. Kegel. State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs). RFC 5128 (Informational), March 2008.
- [44] L. Wang and J. Kangasharju. Real-world sybil attacks in BitTorrent mainline DHT. In *IEEE GLOBECOM*, 2012.
- [45] Z. Wang, Z. Qian, Q. Xu, Z. M. Mao, and M. Zhang. An Untold Story of Middleboxes in Cellular Networks. In *ACM SIGCOMM*, 2011.
- [46] J. Weil, V. Kuarsingh, C. Donley, C. Liljenstolpe, and M. Azinger. IANA-Reserved IPv4 Prefix for Shared Address Space. RFC 6598 (Best Current Practice), April 2012.
- [47] D. Wing. NAT Tutorial. In *IETF 78*, 2010.