# Mass Adoption of NATs: Understanding the Problems and How to Work Around Them

Orestis Kanaris
*Distributed Systems*
*Delft University of Technology*
Delft, Netherlands
o.kanaris@student.tudelft.nl

***Abstract*—In recent times, the prevalence of home NATs and the widespread implementation of CGNATs have posed significant challenges to various applications, particularly those relying on Peer-to-Peer communication. This paper addresses these issues by conducting a thorough review of related literature and exploring potential techniques to mitigate the problems. The literature review focuses on the disruptive effects of home NATs and CGNATs on application performance. Additionally, the study examines existing approaches used to alleviate these disruptions. Furthermore, this paper presents a comprehensive guide on how to puncture a NAT and facilitate direct communication between two peers behind any type of NAT. The techniques outlined in the guide are rigorously tested using a simple application running the IPv8 network overlay, along with their built-in NAT penetration procedures. To evaluate the effectiveness of the proposed techniques, 5G communication is established between two phones using four different Dutch telephone carriers. The results indicate successful cross-connectivity with three out of the four carriers tested, showcasing the practical applicability of the suggested methods.**

***Index Terms*—Network Address Translator, CGNAT, NAT puncturing**

## I. Introduction

Internet connectivity nowadays has become a fundamental necessity. With the recent skyrocket in internet-connected devices, the demand for internet access keeps increasing, leading to the inevitable shortage of IPv4 addresses. This was anticipated since the late 1980s leading to the design of IPv6 [1]. However, with the limited supply of IPv4 addresses and the rate of the rollout of IPv6 being slower than the depletion of IPv4, ISPs have had to find ways to conserve their address space.

Carrier-grade NATs (CGNATs) have emerged as a solution to address this issue. CGNATs allow ISPs to share a single IP address with individual devices on a local network and translate them to a single public IP address for communication with the broader internet.

While CGNAT has been successful in conserving IPv4 addresses, it has several implications, particularly with respect to peer-to-peer (P2P) protocols. P2P protocols rely on direct connections between peers, but with CGNAT, direct connections are not possible as all traffic must be routed through the carrier's NAT device. This can lead to increased latency, slower download speeds and reduced overall performance. According to Yangyang Liu et al., [2] when exchanging files on BitTorrent, peers behind NATs tend not to get favoured thus significantly decreasing their download speed and in some cases even degenerating the download into a client-server interaction [2]. There have been multiple suggested ways of bypassing a NAT in order to establish a direct P2P connection but the most prominent and easiest to implement is NAT puncturing where one can establish a direct channel of communication with another peer given that they can find out the receiver's IP address [3]–[6]. This is also apparent in P2P networks involving mobile phones in cellular networks which are by default behind a NAT [7].

Dutch telecommunication providers have also implemented CGNAT to conserve their IPv4 address space. The Netherlands has a relatively high internet usage [8], leading to a fast shortage of IPv4 addresses. The implementation of CGNAT allows Dutch ISPs to share a single IPv4 address among multiple users, thereby conserving the available address space [9].

However, the implementation of CGNAT has not been without controversies. Privacy advocates argue that CGNAT undermines users' privacy by making it difficult to track individual users' online activity since with CGNAT, all users appear to have the same IP address [10]. Additionally, CGNAT makes it more challenging for users to establish secure and direct connections with other users, potentially exposing their private data to more public networks.

This work examines the need and the technology behind NATs, specifically CGNATs and how they affect P2P applications. Then a simple app which reproduces the NAT penetration algorithms of the main literature developed for the scope of this paper will be evaluated by attempting to communicate between various Android phones which use 5G from different carriers operating in the Netherlands.

The subsequent sections explore the impact of NATs on P2P protocols, techniques for penetrating NATs, and the reproducibility of results obtained from the literature. Additionally, this study examines the legal implications associated with Carrier-Grade NATs and proposes an alternative solution that surpasses the limitations of traditional Carrier-Grade NATs. Finally, a comprehensive discussion brings together the findings and insights gained throughout this research. By delving into these sections, we aim to enhance our understanding of P2P network connections, address the challenges imposed by

NATs, and offer potential solutions for a more efficient and scalable network architecture. A brief overview of all the alternatives and workarounds to the problems introduced by NATs reviewed in this paper can be found in table I.

## II. Peer-to-Peer Network Connections

In the context of this essay, it is crucial to first understand what a UDP session is and what an incoming connection is —in the Peer-to-Peer context. This tutorial provides a concise overview of these concepts, explaining what they entail and how they function within UDP-based peer-to-peer networks.

UDP is a lightweight, connectionless transport protocol within the Internet Protocol (IP) suite. In UDP peer-to-peer connectivity, a UDP session refers to the exchange of data between two peers without the need for a persistent connection. Key characteristics of UDP sessions in peer-to-peer networks are as follows:

`Connectionless Communication`: UDP sessions operate without establishing a dedicated connection between peers. Instead, UDP sends independent datagrams, each containing the necessary information to reach its destination. Peers can initiate data transmission without prior handshaking or negotiation.

`Unreliable Delivery`: Unlike protocols like TCP (Transmission Control Protocol), UDP does not provide built-in error correction or retransmission of lost packets. Consequently, UDP sessions offer unreliable delivery, meaning the protocol does not guarantee that every packet will be received. It is the responsibility of the receiving application to handle any packet loss or errors.

`Datagram Structure`: UDP transmits data in discrete units called datagrams. Each datagram contains a source and destination port number, along with payload data. The size of the payload is limited by the maximum transmission unit (MTU) of the network. Peers can exchange these datagrams freely, allowing for quick and lightweight communication.

**Incoming Connections:** In the context of UDP peer-to-peer connectivity, an incoming connection occurs when one peer establishes communication with another peer by sending a UDP packet. The process of establishing an incoming connection typically involves the following steps:

`Peer Discovery`: Peers within a UDP peer-to-peer network employ various mechanisms to discover and identify each other. This can include techniques such as broadcasting, multicasting, or using a centralized server for peer coordination.

`Packet Exchange`: Once a peer has discovered another peer, it can initiate communication by sending a UDP packet to the target peer's IP address and port number. The packet may contain information about the requesting peer's identity, the desired data, or any other relevant details.

`Response Handling`: The receiving peer processes the incoming packet and formulates an appropriate response. The response may contain the requested data, acknowledgement, or other necessary information. This two-way communication enables the establishment of an incoming connection between the two peers.

UDP peer-to-peer connectivity allows for decentralized and efficient communication between peers, making it suitable for various applications such as file sharing, voice and video streaming, and online gaming.

### A. Network Address Translators

A network Address Translator (NAT) is a piece of hardware or software that holds a table of pairs of local and globally unique IP addresses. Locally IP addresses within the stub domain are not globally unique, so they cannot be used to route packets on the Internet. The packets will have the NATs public IP address and when they are routed to it, the NAT will look up the local IP address that corresponds to the specific IP:Port pair the received packet holds. This method is used to address the problem of IP address depletion by basically "bundling" a LAN behind the NAT box and routing all packets using a single IP address [23]. A potential home NAT setup is depicted in figure 1

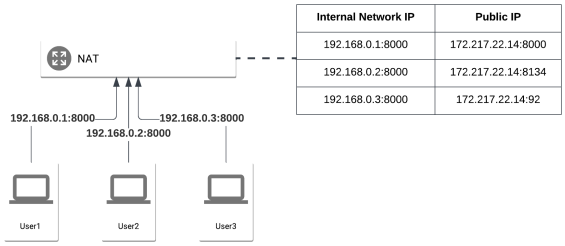


FIG. 1: A NAT setup

However, it may not be a suitable long-term solution and could also lead to short-term problems. NATs face scalability issues since as tables scale there will be an overhead on the table lookup. Dynamically allocating IP:Port pairs also increase the probability of misaddressing since a delayed packet may end up on a collision since the NAT may have already re-allocated that pair. A crucial problem with NATs is that some applications cannot function properly or they break. Examples of these applications are video streaming, online games and P2P-dependent applications. [23].

According to the STUN protocol [11], there are four types of NATs, namely Full-cone NAT, Restricted-cone NAT, Port-restricted cone NAT and Symmetric NAT. These types fall into two categories according to RFC4787 [24] namely the "easy" NATs which do Endpoint-Independent Mapping (EIM) and the "hard" NATs which do Endpoint-Dependent Mapping. EIM ensures the consistency of the external address and port pair if the request is coming from the same internal port.

According to Huawei [25] the specifications of these types of NATs are:

**Full-cone NAT** An `EIM` NAT where all requests coming from the same internal IP1:Port1 pair are mapped to the same public IP2:Port2 pair. On top of that, any host on the Internet

| Technique | Description | Literature |
|---|---|---|
| NAT Punctuting | A technique that allows direct communication between two devices behind separate NAT routers by utilizing temporary port mappings or establishing a relay server to facilitate communication. | [3]–[6], [11] |
| Birthday-Paradox based NAT Puncturing | NAT Puncturing but exploits the birthday paradox to figure out the address port mapping on hard NATs | [12]–[15] |
| QUIC Penetration | If TCP is necessary due to a stream-oriented connection, switch to QUIC and then adapt the UDP-puncturing techniques to work with QUIC | [16] |
| Dual Stack-Lite | An IPv6 transition mechanism that enables the coexistence of IPv4 and IPv6 by encapsulating IPv4 packets within IPv6, allowing service providers to conserve IPv4 addresses while migrating to IPv6 | [17] |
| Extending IPv4 Address Space | Extending the IPv4 address space by "stealling" 6 bits from the port number in TCP/UDP thus extending the address space by 10 bits | [18] |
| Universal Plug'n'Play Internet Gateway Device (UPnP IGD) | Enables devices on a network to automatically set up and manage port forwarding, allowing for seamless communication and connectivity. | [19] |
| NAT Port Mapping Protocol (NAT-PMP) | Simplifies the process of port mapping by allowing devices to automatically request and configure port mappings on a network's NAT router. | [20] |
| Port Control Protocol (PCP) | Similar to NAT-PMP but supports IPv6 too | [21] |
| Interactive Connectivity Establishment (ICE) | Facilitates the establishment of peer-to-peer connections across networks by dynamically selecting and negotiating the optimal communication path | [22] |

TABLE I: Overview of all peer-to-peer techniques to establish communication behind NATs

can communicate with the host on the LAN by sending packets to the mapped public IP address and port.

**Restricted-cone NAT** An `EIM` NAT where similar to the Full-cone NAT an internal IP:Port pair will be mapped to the same external IP:Port pair. The difference with this NAT is that a host on the Internet can send packets to a machine behind the NAT only if that machine initiates the communication.

**Port-restricted cone NAT** An `EIM` NAT, similar to the Restricted-cone NAT but the restriction includes port numbers.

**Symmetric NAT** An `EDM` NAT where all requests coming from the same internal IP1:Port1 pair are mapped to the same public IP2:Port2 pair. The difference with this NAT is that it also takes into account the destination of the packet i.e a request from internal IP1:Port1 to external IP2:Port2 will have a different mapping from a request IP1:Port1 to external IP3:Port3 thus two consecutive requests from the same internal pair but to different external hosts will lead to two different mappings.

### *B. Carrier-Grade NATs*

While hiding local home networks behind NAT boxes did help with alleviating the problem of depletion of addresses for a while; this was not enough. Many ISPs worldwide are running out of IP addresses to allocate, thus they resulted in bundling different customers and areas together behind a NAT box i.e. a Carrier-Grade NAT (CGNAT). ISPs worldwide started rolling out CGNATs but by 2016 it had received very little empirical assessment [26], [27]. Some general problems of the kind that the everyday user might encounter, were identified in a test performed on a Turkish ISP [28]. These are:

- Users are unable to access remote desktops or cameras
- Users cannot open a port on demand or cannot access it if it is already opened.
- Being in a weak CGNAT IP pool may affect the user's connection speed and it can also result in high ping
- Latency issues can occur due to the extra hop node
- Issues can occur with services allowing registration or log in from only one IP in the case multiple users behind the same CGNAT attempt to access it (a testimony of this also exists on a Ziggo forum from a user who complained about not being able to access google.com or any META-owned site [29])

An assessment contributed to the RFC series of the Internet Engineering Task Force (IETF) identified various services where a CGNAT may cause them to break or degrade their performance [17]. In general, the testing revealed that applications such as video streaming, video gaming and P2P file sharing are impacted by CGNAT.

The services that broke are [17]:

- Several P2P applications like XBOX P2P gaming and SIP call using PJSIP client, failed in both the NAT444 and Dual-Stack Lite environments (PJSIP worked when clients used a registration server to initiate calls, given that the client inside the CGNAT initiated the traffic. FTP sessions to servers located behind two layers of NAT failed. When the CGNAT was bypassed and traffic only

needed to flow through one layer of NAT, clients were able to connect).
- Applications that did not first send outgoing traffic thus not opening an incoming port through the CGNAT hence being unable to launch
- Applications that tried to open a particular fixed port through the CGNAT, which works for a single subscriber but not when multiple subscribers try to use the same application.
- Multicast traffic was not able to flow through the CGNAT.

The services whose performance was impacted are [17]:

- Large file transfers initiated on the same (home) network
- Multiple video streaming sessions initiated in the same (home) network
- Sometimes video streaming like Silverlight and Netflix would exhibit a slowdown in single sessions as soon as a second session was established (router dependent issue) —routers that support DSLite did not slow down on single-session video streaming when a second is established

RFC7021 [17] identified some additional problems, which are:

- Loss of geolocation information, something that mainly affects applications that require the precise location of the user and not an approximation of it, since the exact location becomes impossible to get and the only available one might be the location of the CGNAT box.
- Lawful Intercept/Abuse Response, explained in section VI
- Harder for security testers to launch anti-spoofing attacks.

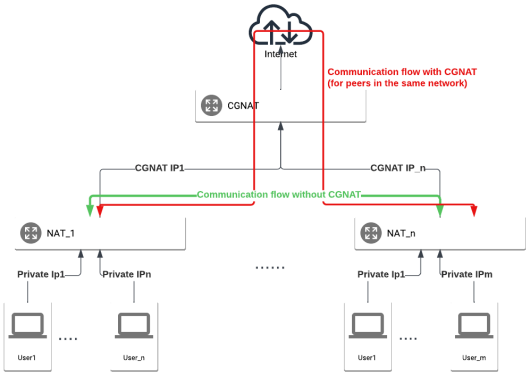


FIG. 2: Communication flow behind a CGNAT

Much research was performed on the problems that NATs and CGNATs cause. For example, the thesis of Fredrik Thernelius [30] identified problems occurring with the Session Initiation Protocol (SIP) when the session is not initiated from the inside of the NAT since otherwise extra steps will be required to find the address to the right internal host. While Victor Paulsamy et al. [31] identified problems with NAT Firewalls when the Voice over Internet Protocol (VoIP) is used.

CGNATs often fail on peer-to-peer communication from users behind the same CGNAT as explained in section IV. The flow of communication is shown in figure 2 with green being the intended communication flow and red being the actual communication flow.

## III. Impact of NATs on P2P protocols

The main interest of this study is how NATs, specifically CGNATs are affecting P2P communications since the majority of customers are protected by several levels of NAT, while data-centre nodes may be hidden behind NAT for the sake of security or virtualization. The utilization of containerized deployments is exacerbating the situation since every communication between peers necessitates a mechanism to navigate NATs, or else operations will be impacted [5]. This problem comes in two-fold since NATs sometimes act as Firewalls meant to block incoming traffic from entering the LAN unless those packets are a response to a communication established from the inside.

J.J.D Mol et al. [32] established in their 2008 research on fairness for BitTorrent users chain, that peers that are behind firewalls have more difficulty obtaining a fair sharing ratio; thus they concluded the need for puncturing NAT or using a static IP address in order to optimize the performance of the network.

This problem of clients behind NATs being unfavored by the BitTorrent protocol can be highly observed in networks like Tribler — a BitTorrent [33] based open source project that extends the protocol by adding features such as video on demand and live streaming while remaining fully backwards compatible [34]— because the vast majority of the peers are mobile devices. The vast majority of mobile phones are behind NATs since there is no option for a static IP address on a mobile device, and cellular networks use carrier-grade NATs.

As noted by J. Pouwelse et al. [35], most consumers are behind a number of layers of NATs thus the new implementations/versions of P2P-based networks should take this into account. According to the same authors, in order to build effective P2P networks, the network designers need to consider that many users are behind non-specially configured NATs and firewalls in their home setups (potentially also CGNATs). To date, no elegant solution is found for P2P TCP-based communication through NAT/firewall, which means that the network needs to be UDP-based to allow for NAT traversals.

## IV. Penetrating a NAT

As established in Section III, most users will have some troubles with P2P communication when they reside behind a NAT and/or a firewall since P2P networks assume that all nodes in the network are connectable. For UDP-based protocols, the UDP hole-punching technique has been proposed to overcome this problem. The idea is since the NAT will block packets that are incoming from connections that have not been mapped yet, then the internal endpoint must send some packet first to the external remote endpoint thus creating a “hole“ in the NAT or firewall through which then communication

can then proceed [3]–[5]. Note that TCP puncturing is also possible but adds extra levels of complexity thus if TCP is necessary due to a stream-oriented connection, the user should consider switching to QUIC and then adapt the UDP-puncturing techniques to work with QUIC [16]. Different steps (which will be explained in this chapter) are required based on the type and amount of NATs/Firewalls and whether one or both users are behind a NAT/Firewall.

As mentioned in Section II-A, NATs are divided into “easy“ and “hard“ with easy being the ones which are relatively easier to penetrate. The algorithm to penetrate NATs using the Birthday Paradox [12] is derived from a blog post series by David Anderson [13], [14]. Note that throughout this text the terms NAT traversal, NAT penetration, puncturing, and UDP hole-punching will be used interchangeably.

Before going to the algorithm, it’s good to note that most NATs include a stateful firewall which is a piece of software/hardware that recalls the packets previously encountered and applies that information when determining how to handle new packets that arrive.

Starting off, to perform NAT traversal there are two requirements, that is *the communication protocol between the two peers need to be UDP based* and *the peer(s) behind the NAT needs to have direct control over the network socket that is sending and receiving network packets*. To bypass the stateful firewall using UDP is straightforward: the firewall permits an incoming UDP packet only if it has previously detected a corresponding outbound packet. Hence the computer located behind the firewall must be the initiator of the connections. However, a problem arises when two peers located behind firewalls wish to communicate directly (since the firewalls are now ”facing each other”). No one can make the first move because each side is waiting for the other to take the initiative.

The solution to the double firewall problem comes from the observation that the firewall rule says that packets must flow out before packets can flow in, `but they don't necessarily have to be related to each other`. As long as the incoming packet has the expected source and destination then any packet can be a response to the outgoing one. Thus to traverse multiple stateful firewalls, there needs to be some information shared in advance, i.e. the IP:port that each peer is using, which can be either manually configured — something that does not scale quickly— or the peers can use a coordination server to keep the IP:port information synchronized in a secure and flexible way as in figure **??**. Note that this technique requires precise timing, roughly the second step (of initiating the puncture), needs to be done at the same time thus a clock should be used.

There are three possible scenarios, both peers being behind EIM-based NATs, one being behind an EIM-based NAT and the other behind an EDM-based NAT or both peers being behind EDM-based NATs.

### A. Peers that are both behind an EIM-based NAT

Now when both peers are behind an EIM-based NAT things get harder, since the peers don’t know their external

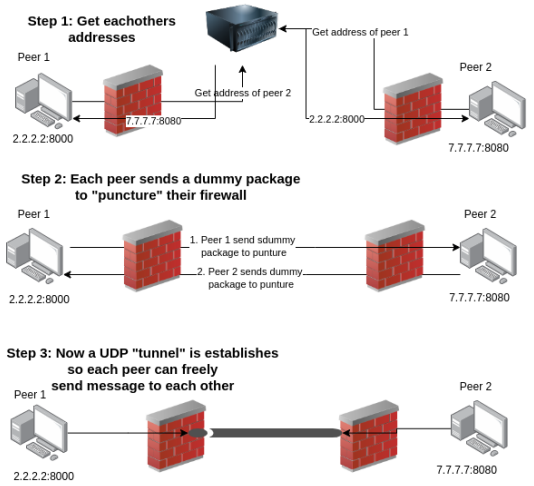


FIG. 3: Two machines behind Firewalls using a synchronizer server to find each other’s IP address and then puncture their firewalls to establish a communication channel between each other

IP addresses —strictly speaking, there is no external IP:port until the other peer sends packets since NAT mappings are only created when outbound packets are required to flow towards the Internet. Basically, both need to initiate the communication first, but no one knows who to send the packet to. This problem of one not knowing their IP address is what the Session Traversal Utilities for NAT (STUN) is aiming to solve. The idea behind it is that when a client behind a NAT communicates with a server on the Internet, the server sees the public IP address of the client, not the intranet one. Thus the server can reply to the client with the IP:port that the server saw when it received the message [11]. Note that depending on the NAT type, the IP:port that STUN “sees“ is not always the same as the one the whole Internet sees. EDM-based NATs as explained in section II-A create a different mapping for every single destination.

### B. Peers where one is behind an EIM-based NAT and the other behind an EDM-based NAT

If one peer is behind an EDM-based NAT that means that the NAT is opening a different port for each communication. Since there are 65535 different ports [36], if the peer behind the EIM-based port attempted to brute force it, assuming median Internet speed in the Netherlands which at the time of writing is $\approx$ 128 Mbps download and $\approx$ 40 MBps upload [37] the maximum packets per second that a machine can send is $\approx$ 56000, meaning that it can brute force the NAT mapping in little over 1s.

Problems arise when the Internet upload speed is low, the NAT/Firewall has some rules to block brute-forcing or both of the peers are behind an EDM-based NAT.

### C. Peers where both are behind an EDM-based NAT

Assuming that someone would try to brute force between two peers that are both behind EDM-based NATs. Since the initiator of the brute force attack does not know their own mapping either, this means that now there are $65535^2 = 4294836225$ possible combinations. Assuming the average Internet connection in the Netherlands, a brute-force attack would take roughly 21 hours.

To improve on this, one can use the Birthday Paradox to achieve a collision in significantly less time.

The **Birthday Paradox** is a problem in probability theory that involves determining the likelihood of at least two individuals sharing the same birthday among a group of n randomly selected people. Surprisingly, the paradox reveals that only 23 people are needed to reach a 50% probability of shared birthdays. This may seem counter-intuitive, but the reasoning behind this is that every possible pair of individuals within the group will be compared. Therefore, with 253 pairs to consider (calculated by $\frac{23 \cdot 22}{2}$), which is more than half the number of days in a year, it becomes easier to understand why this result holds true [12]. This paradox also has practical applications, such as the "birthday attack," which uses this probabilistic model to reduce the complexity of finding a collision. In this case trying to find a collision of two pairs of IP:port.

From the calculations performed in [13] one can get a 50% success rate of double IP:port collision after sending $\approx$ 54000 packets, something that can be achieved in under a second, assuming an average Netherlands connection. To achieve a 99.9% success rate $\approx$ 170000 packets are needed, where assuming the same connection it can be achieved in little over 3 seconds. This is a tremendous improvement from the 21 hours needed without using a birthday attack. Note that a birthday attack could also be used in the scenario of one peer behind an EIM-based NAT and the other behind an EDM-based NAT to achieve an almost instant collision with 99.9% probability.

The code associated with the algorithm above can be found on this GitHub page [15].

### D. Port Mapping Protocols

There exist three protocols for partially manipulating port maps. Something like asking the NATs to allow more stuff in, so we don't have to result in brute force or birthday attacks. There are three protocols to do this and they will be explained in this subsection.

**Universal Plug'n'Play Internet Gateway Device (UPnP IGD)** is the oldest of the three, developed in the late 1990s and uses old technologies like XML, SOAP and multi-cast HTTP over UDP. It allows the user to perform a request to map ports in their —-UPnP IGD-enabled — NAT. One should keep in mind that is hard to implement correctly and securely [19].

**NAT Port Mapping Protocol (NAT-PMP)**, a protocol developed by Apple as a competitor of UPnP IGD was designed to only perform port forwarding while being much easier to implement both on clients and on NAT devices [20].

**Port Control Protocol (PCP)** also known as NAT-PMPv2 is similar to NAT-PMP but also support IPv6 [21].

Having these protocols in mind one can try to look up any of these three protocols on their local default gateway and request a public port mapping thus further simplifying the connectivity between hosts behind NATs; although it is common that these protocols are disabled.

To decide on what technique to use at any time, be it brute force, birthday paradox, port mapping etc there is the Interactive Connectivity Establishment (ICE) developed in 2010 [22].

**Interactive Connectivity Establishment (ICE)** simply put is an algorithm where one tries every technique at once and picks the best technique that works. According to IETF, the ICE protocol is a NAT traversal technique, a multi-homed address selection technique and a dual-stack address selection technique that works by including multiple IP addresses and ports in both the request and response messages of a connectivity establishment transaction without making any assumption about the network topology [22].

### E. CGNATs

Before the rollout of CGNATs, users could bypass their NATs by using any port mapping protocol to configure port forwarding on their home routers. Unfortunately, the ISPs' CGNATs are not reconfigurable.

Fortunately, though the existence of a CGNAT throughout the routing path will not require any significant changes since it is practically a double NAT so the same algorithm can be used since the only NAT that should concern the user is the last one from the Internet. Still, one should expect that more time will be required to crack them since there will be significantly more combinations to test.

A problem arises when both peers are behind the same CGNAT. This is a problem since STUN won't work since the STUN server will be outside of the intranet and will see the "middle" network — since CGNAT effectively develops a "mini" Internet for the devices connected to it—every time a "what is my address" request is performed.

In case hairpinning is supported by the CGNAT we can try to use that. Hairpinning is when both internal peers use a STUN server to get their external IP address and then the other peer just sends the packets to that IP address hoping that they will go through. Though it is not always the case that hairpinning is supported by the CGNATs [24].

In the case that hairpinning fails, then the user's only option left is relaying.

## V. Reproducing results from literature

A simple app running IPv8[1] was developed in order to test the penetration rate of NATs on mobile on various Dutch

[1] https://github.com/Tribler/kotlin-ipv8

| | T-Mobile | Lebara | Lyca Mobile | Vodafone |
|---|---|---|---|---|
| T-Mobile | ✓ | ✓ | × | ✓ |
| Lebara | ✓ | ✓ | × | ✓ |
| Lyca | × | × | × | × |
| Vodafone | ✓ | ✓ | × | ✓ |

TABLE II: Carriers that succeeded in transmitting a package to another carrier

telecom providers. Four different SIM cards were used i.e. Lyca[2], Lebara[3], T-Mobile[4] and Vodafone[5].

The goal of the app was to determine whether IPv8 could make two phones each on a different carrier's 5G running kotlin-ipv8 and a computer running on WiFi the JVM version of IPv8 discover each other and communicate by penetrating potential NATs that are in the way as can be seen in figure 8. The experiment failed on Lyca since IPv8 was unable to penetrate the Symmetric NAT that Lyca has in place as can be seen in figure 4.

The other 3 carriers achieved communication both with each other and with the computer (as can be seen in table II and figure 5), with IPv8 reporting that it managed to exchange the preset UDP messages between the devices while also penetrating the NATs that were in place and keeping the hole alive throughout the whole duration of the experiment. This represents a 75% success rate of NAT penetration across the four different Dutch carriers tested.

There was no packet loss observed between the three carriers, but there were some instances where the received packets had invalid public keys, thus they could not be decoded.

## VI. Legal Implications of Carrier-Grade NATs

As already discussed, CGNATs may crush or slow down some applications, but they may impact law enforcement investigations into online crime.

Part of these investigations is to gather intelligence which requires knowledge of Internet communications. This was much easier before NATs since ISPs would just store a table of IP addresses, who used them and the time period of that. ISPs are generally required to store this information under the "data retention" law.

With the rollout of CGNATs, this format of data retention is not sufficient anymore since one does not have their own IP address allocated. The new format should be, the IP address allocated, the IP address used and the port number along with a timestamp, but this is very inefficient due to the huge amount of data that this format will generate. Assuming a large ISP of 25 million customers, and users averaging 33 thousand connections a day, this averages to a log file of about 425TBytes. This is highly inefficient and significantly slow to query [38].

[2]https://lycamobile.nl/en/
[3]https://mobile.lebara.com/nl/en
[4]https://www.t-mobile.nl/
[5]https://www.vodafone.nl/

Politicians, law enforcement agencies and ISPs are collaborating to format the new laws around data retention to allow law enforcement to find the perpetrator of a crime while not overloading the ISPs' data centres. This should be done with great care since when multiple people are sharing a single IP address when a crime is committed there should be a way to identify the exact source of the crime to avoid wrongly investigating innocent individuals who just happen to share an IP address with a criminal [10].

## VII. A better Approach than Carrier-Grade-NAT

The most obvious solution to removing CGNATs is a full rollout of IPv6, completely replacing IPv4. Although this is the theoretical eventual goal, it won't be realized soon. As of April $23^{rd}$ 2023 — 11 years after the launch of IPv6—, google statistics show 41.93% of their users having adopted IPv6 [39] thus it is fair to assume that there is a long way until a complete IPv6 adoption by the Internet.

Since there is no specific day that IPv4 support will come to an end and a full transition to IPv6 will be rolled out, some nodes on the internet are currently only running on IPv4 and some on IPv6. A core value of the internet is that any node can communicate with any other node, thus a seamless translation mechanism is required between these nodes [40].

Enter NAT46 and NAT64. These are network address translators but their purpose is to translate from IPv4 to IPv6 and vice-versa. The need for these NATs is currently unavoidable at this state but it also introduces the same problems as regular NATs.

Until IPv6 is completely rolled out, different researchers have proposed solutions to depleting IPv4 addresses that do not involve CGNATs. One of these suggestions came from O. Maennel et al. [18] where they suggested extending the IPv4 address space by "stealling" bits from the port number in TCP/UDP —thus still allowing users to utilize a single IPv4 address.

Their suggestion called extended addressing, is to limit the port addressing to 6 bits, an action that will increase the address space by 10 bits, thus multiplexing 1024 additional users per existing IP address.

This suggestion will reduce the number of fixed range of ports that applications can use. Still, the need for IP addresses is stronger than the need to run thousands of applications on a single host, something that broadband consumers are not anticipated to do.

## VIII. Discussion

This paper highlights the issues associated with the wide adoption of NATs and particularly CGNATs and how they affect different applications while also raising legal and societal issues. The ideal solution is full adoption of IPv6, which is not coming anytime soon thus NATs are unlikely to go away any time soon.

Multiple different solutions were presented in the paper, including NAT puncturing in order to allow the smooth working of peer-to-peer protocols to an adaptation of IPv4 where fewer


FIG. 4: IPv8 reporting failure on penetrating Lyca due to Symmetric NAT

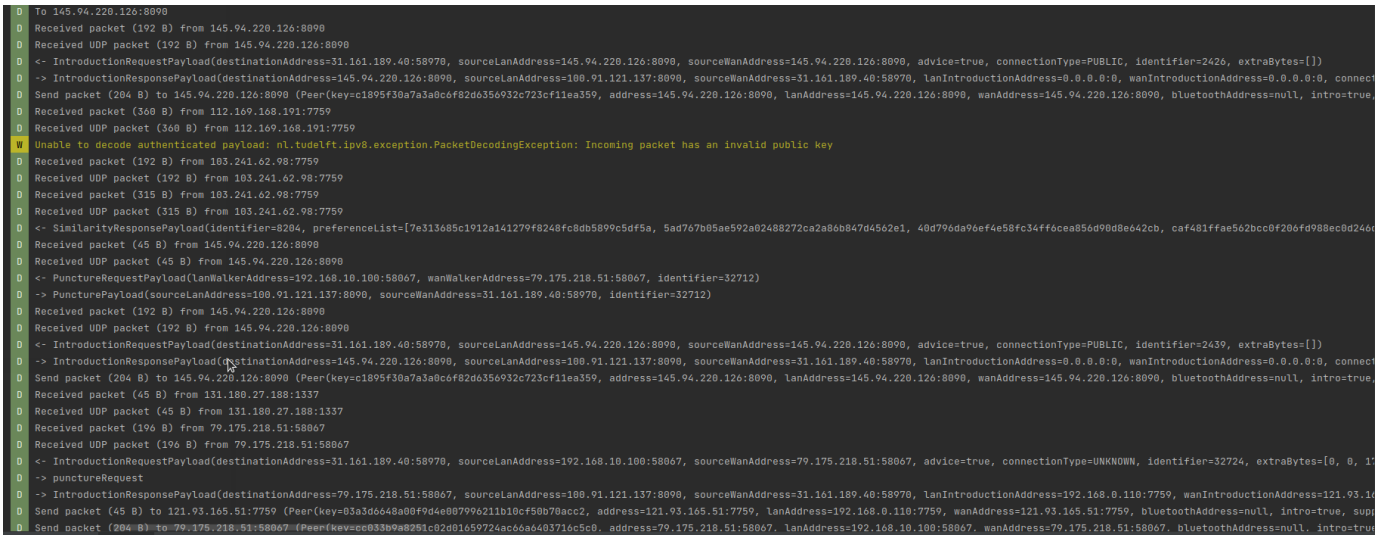


FIG. 5: Lebara successfully communicating with Lebara and the JVM

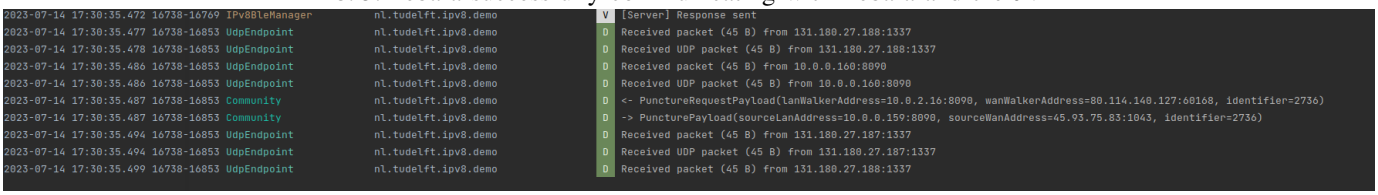


FIG. 6: Logs of the successful puncturing

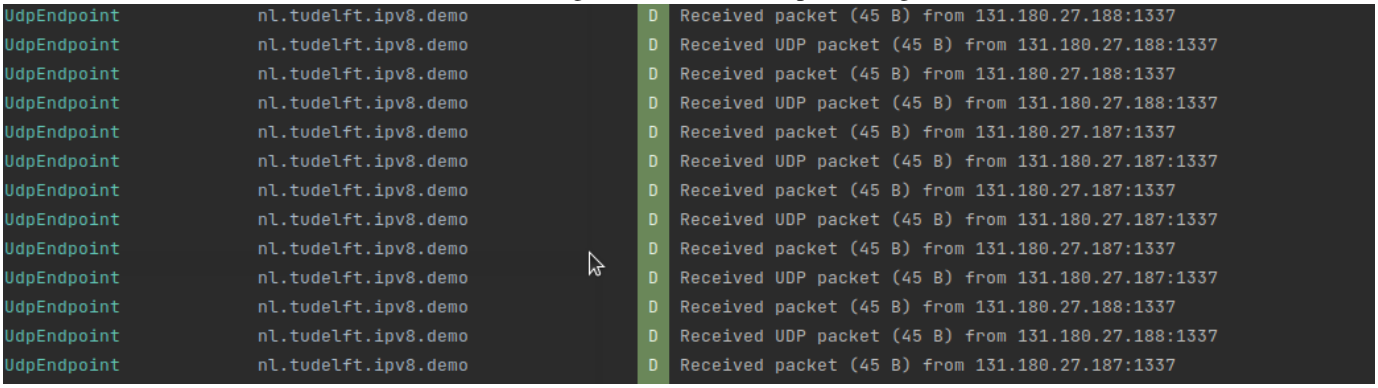


FIG. 7: Packets send and successfully received

bits from the port number will be used to extend the address part of IPv4, thus increasing the available address space. Each solution is good for different scenarios and it also depends on whether the user is able to implement these techniques.

Fortunately, according to Bryan Ford et al. [4], as NAT vendors become more aware of the requirements of significant P2P applications like Voice-over-Internet protocol and online gaming protocols, they will probably enhance their support for hole punching in the coming times.

## REFERENCES

[1] N. R. Murphy and D. Malone, *IPv6 Network Administration: Teaching the Turtle to Dance*. " O'Reilly Media, Inc.", 2005.

[2] Y. Liu, L. Chang, and J. Pan, "On the performance and fairness of bittorrent-like data swarming systems with nat devices," *Computer Networks*, vol. 59, pp. 197–212, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128613003770

[3] G. Halkes and J. Pouwelse, *UDP NAT and Firewall Puncturing in the Wild*. STACS 98, 2011, p. 1–12.

[4] B. Ford, P. Srisuresh, and D. Kegel, "Peer-to-peer communication across network address translators." in *USENIX Annual Technical Conference, General Track*, 2005, pp. 179–192.

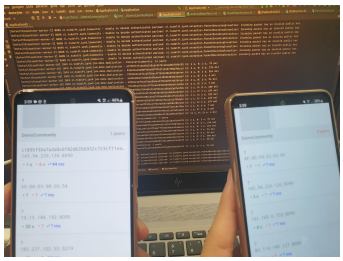

FIG. 8: The setup used


FIG. 9: The sims used for testing


[5] J. Pouwelse, "Trustchain protocol," Internet Engineering Task Force, Internet-Draft draft-pouwelse-trustchain-01, Jun. 2018, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-pouwelse-trustchain/01/

[6] S. Guha and P. Francis, "Characterization and measurement of tcp traversal through nats and firewalls," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, 2005, pp. 18–18.

[7] W.-K. Jia, "Pfqdn: Sdn- and dns-assisted transparent communications among behind-nat networks," *IEEE Systems Journal*, vol. 17, no. 2, pp. 2271–2281, 2023.

[8] S. Kemp, "Digital 2022: The netherlands - datareportal – global digital insights," Feb 2022. [Online]. Available: https://datareportal.com/reports/digital-2022-netherlands

[9] M. Skála, "Technology stack for decentralized mobile services," 2020.

[10] "Are you sharing the same ip address as a criminal? law enforcement call for the end of carrier grade nat (cgn) to increase accountability online," Oct 2017. [Online]. Available: https://www.europol.europa.eu/media-press/newsroom/news/are-you-sharing-same-ip-address-criminal-law-enforcement-call-for-end-of-carrier-grade-nat-cgn-to-increase-accountability-online

[11] J. Rosenberg, C. Huitema, R. Mahy, and J. Weinberger, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," RFC 3489, Mar. 2003. [Online]. Available: https://www.rfc-editor.org/info/rfc3489

[12] C. Swadling, "The birthday paradox," *Leicester Undergraduate Mathematical Journal*, vol. 1, 2019.

[13] D. Anderson, "How nat traversal works," Aug 2020. [Online]. Available: https://tailscale.com/blog/how-nat-traversal-works/

[14] ——, "How nat traversal works - nat notes for nerds," Apr 2022. [Online]. Available: https://blog.apnic.net/2022/04/26/how-nat-traversal-works-nat-notes-for-nerds/

[15] Danderson, "Nat-birthday-paradox/paradox.py at master · danderson/nat-birthday-paradox." [Online]. Available: https://github.com/danderson/nat-birthday-paradox/blob/master/paradox.py

[16] K. Y. Gbur and F. Tschorsch, "A QUIC(K) way through your firewall?" *CoRR*, vol. abs/2107.05939, 2021. [Online]. Available: https://arxiv.org/abs/2107.05939

[17] C. Donley, L. Howard, V. Kuarsingh, J. Berg, and J. Doshi, "Assessing the Impact of Carrier-Grade NAT on Network Applications," RFC 7021, Sep. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc7021

[18] O. Maennel, R. Bush, L. Cittadini, and S. M. Bellovin, "A better approach than carrier-grade-nat," 2008.

[19] 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, Jul. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6970

[20] S. Cheshire and M. Krochmal, "NAT Port Mapping Protocol (NAT-PMP)," RFC 6886, Apr. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6886

[21] D. Wing, S. Cheshire, M. Boucadair, R. Penno, and P. Selkirk, "Port Control Protocol (PCP)," RFC 6887, Apr. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6887

[22] A. Keränen, C. Holmberg, and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal," RFC 8445, Jul. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8445

[23] K. Egevang and P. Francis, *The IP Network Address Translator (NAT)*, 1994.

[24] C. F. Jennings and F. Audet, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP," RFC 4787, Jan. 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc4787

[25] L. Qiaoqiao, "What is nat? what are the nat types?" Sep 2021. [Online]. Available: https://info.support.huawei.com/info-finder/encyclopedia/en/NAT.html

[26] I. Livadariu, K. Benson, A. Elmokashfi, A. Dhamdhere, and A. Dainotti, "Inferring carrier-grade nat deployment in the wild," 2018.

[27] [Online]. Available: https://techhub.hpe.com/eginfolib/networking/docs/routers/hsr6800/5200-3510_l3-ip-svcs_cg/content/482615968.html

[28] K. O. Akpinar, M. Akpinar, I. Ozcelik, and N. Yumusak, "Carrier-grade nat — is it really secure for customers? a test on a turkish service provider," 2016.

[29] door Sander Community Developer and C. Developer, "Cgnat zorgt voor problemen?" Jan 2023. [Online]. Available: https://community.ziggo.nl/t5/Archief/CGNAT-zorgt-voor-problemen/td-p/921374

[30] F. Thernelius, "Sip, nat and firewalls," *Kungl Tekniska Hogskolan*, May 2000. [Online]. Available: http://www.recursosvoip.com/docs/english/Ther0005_SIP.pdf

[31] V. Paulsamy and S. Chatterjee, "Network convergence and the nat/firewall problems," 2003.

[32] J. Mol, J. Pouwelse, D. Epema, and H. Sips, "Free-riding, fairness, and firewalls in p2p file-sharing," 2008.

[33] B. Cohen, "Bittorrent.org," Feb 2017. [Online]. Available: http://www.bittorrent.org/beps/bep_0003.html

[34] N. Zeilemaker, M. Capotă, A. Bakker, and J. Pouwelse, "Tribler: P2p media search and sharing," in *Proceedings of the 19th ACM International Conference on Multimedia*, ser. MM '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 739–742. [Online]. Available: https://doi.org/10.1145/2072298.2072433

[35] L. D'Acunto, J. Pouwelse, and H. Sips, "A measurement of nat & firewall characteristics in peer to peer systems," *Proc. 15-th ASCI Conference*, vol. 5031, 01 2009.

[36] "What is a computer port? — ports in networking — cloudflare." [Online]. Available: https://www.cloudflare.com/learning/network-layer/what-is-a-computer-port/

[37] "Netherlands's mobile and broadband internet speeds." [Online]. Available: https://www.speedtest.net/global-index/netherlands#fixed

[38] G. Huston, "The isp column - dotnxdomain.net," May 2013. [Online]. Available: https://www.dotnxdomain.net/ispcol/2013-09/valuation.pdf

[39] "Ipv6 statistics google," Apr 2023. [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html

[40] A. Durand, "NAT64 - NAT46," Internet Engineering Task Force, Internet-Draft draft-durand-ngtrans-nat64-nat46-00, Jun. 2002, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-durand-ngtrans-nat64-nat46/00/