Edge-Case Integration into Established NAT Traversal Techniques

Simon Keller
University of Wuerzburg
Department of Computer Science
simon.keller@stud-mail.uni-wuerzburg.de

Tobias Hoßfeld
University of Wuerzburg
Department of Computer Science
tobias.hossfeld@uni-wuerzburg.de

Sebastian von Mammen
University of Wuerzburg
Department of Computer Science
sebastian.von.mammen@uni-wuerzburg.de

2022 IEEE Ninth International Conference on Communications and Electronics (ICCE) | 978-1-6654-9745-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCE55644.2022.9852092

***Abstract*—Traversing Network Address Translation (NAT) is often necessary for establishing direct communication between clients. The traversal of NAT with static port translation is solved in many cases by the Session Traversal Utilities for NAT (STUN) protocol. Nevertheless, it does not cover the traversal of progressing symmetric and random symmetric NAT, which make it necessary to correctly predict opened ports. This paper presents a method for predicting (a) progressing symmetric NAT-translated ports based on a network traffic model and the Expected Value Method, and (b) random symmetric NAT-translated ports based on heuristics between monitored and opened ports across numerous traversal attempts. Tests were conducted in German cities using local cellular communication providers. Compared to established approaches, they yielded considerable improvements traversing progressing symmetric NAT and slight improvements traversing random symmetric NAT in real-world environments.**

I. Introduction

In a private network, each host can be assigned a private IP address. If a host is accessing the internet via a Network Address Translation (NAT) device, the private IP address is translated to a public IP address. Multiple devices with different private IP addresses can use a single public IP address. This process was originally invented to slow down the depletion of IPv4 addresses as well as for security reasons [2]. Today, most private connections rely on NAT, which requires NAT traversal for most direct connections [15]. Due to a lack of NAT standardization, numerous implementations emerged, which is why the traversal process can be more or less challenging. There are solutions for traversing NAT, when the translation of private IP addresses to public IP addresses is static, which is broadly implemented in home routers [6]. Unfortunately, there are no established standards for symmetric NAT translated IP addresses. NAT implementations of this kind are heavily used in public facilities and mobile networks. Most of the current solutions for symmetric NAT situations are based on forwarding communication packets using relay servers. These methods may suffer from increased latency, bandwidth limitations and extra costs for servers which depends on the actual application. Müller, Klenk, and Carle [7] discovered that the requirements of NAT traversal for legacy applications differ significantly yielding to various service categories relevant for NAT traversal. Decentralized approaches and in particular structured P2P overlay network may overcome the NAT-traversal problem to connect nodes that are not directly addressable over the Internet. However, this requires extra efforts in terms of signalling traffic and bandwidth overhead [3, 10, 12]. This paper proposes a new lightweight method for establishing direct communication between peers in symmetric NAT situations based on port predictions. We answer the following research questions: *Can we improve the ratio of successful NAT traversals in the presence of a) progressing and b) random symmetric NAT in real-world environments compared to the state-of-the-art?* The key contribution of this work is a) refinement and combination of existing NAT traversal approaches, b) an implementation and measurement study on establishing direct communications between clients in public mobile and fixed networks in Germany.

In the following section, background information and established NAT traversal techniques are presented. Section 3 details the proposed NAT traversal procedure, followed by the port prediction algorithms for symmetric NAT in Section 4. Experimental results generated from the proposed approaches are presented in Section 5. Finally, conclusions are drawn in Section 6.

II. Background and Related Work

The success rate of established NAT traversal techniques strongly depends on the type of NAT each peer is connected to. The three general types of NAT are (1) full Cone, (2) restricted Cone and (3) symmetric. (1) In full Cone NAT, each private address and port is mapped statically to a public address and port. The port translation is predefined for any clients behind the NAT device. There are no restrictions on incoming packets [1]. (2) Restricted Cone NAT can be divided into address restricted Cone NAT and port restricted Cone NAT. Different to full Cone NAT, address restricted Cone NAT only allows incoming packets from external hosts, if the client has previously sent a packet to this host. In addition, port restricted Cone NAT only forwards packets from external hosts, if they come from the same address and port number the client has previously sent a packet to. (3) Symmetric NAT implementations can also be divided into two different types, i.e. progressing symmetric and random symmetric. In contrast to (1) and (2), symmetric NAT uses a new port mapping when the requested target address changes. Any request from an internal address and port to some destination address and port is mapped to a unique public address and port [13]. *Progressing symmetric NAT* means to assign the mapping ports with continuous and progressive numbers, i.e., the public ports are translated by the symmetric NAT device in a progressing sequence. In *random symmetric NAT*, private ports are mapped pseudo-randomly to some or the full port range.

978-1-6654-9745-9/22/$31.00 ©2022 IEEE

A field study [7] investigates the success rates of promising NAT-Traversal techniques deployed in the wilds. This study from 2008 shows that 85% of all NATs were either of type Port-Address Restricted or Full-Cone. Symmetric NATs were rarely discovered ($< 5\%$). The traditional protocols for traversing NAT are STUN [9], Traversal Using Relays around NAT [6] (TURN) and Interactive Connectivity Establishment [8] (ICE). While STUN establishes direct communication between peers in non-symmetric NAT situations, TURN and ICE are using relaying solutions to traverse symmetric NAT devices. Relaying solutions have several drawbacks like bandwidth limitations, increased latency and server costs that scale with the number of users behind symmetric NAT.

A similar approach to STUN is called hole-punching. There, the public address and port is exchanged between both peers. After the address information is exchanged, one peer is punching a hole using low time to live (TTL) values and the second peer is targeting the hole with a regular packet. Unfortunately, hole-punching is, by default, unable to traverse symmetric NAT, since in this case a new mapping is used when both peers are sending their traversal packets.

Wei et al. [13] present traversal methods for symmetric NAT based on a NAT identification procedure and hole-punching. The traversal process is divided into three phases–first NAT type identification of each peer, second prediction of the next allocated port of the NAT devices and third the traversal process. A similar structure is also proposed in this paper and presented in the following section. The port prediction procedure for progressing symmetric NAT from Wei et al. [13] works as follows. When the source port numbers are $\{x, x+1, x+2, ...\}$ and the predicted translated port numbers are $\{n, n+1, n+2, ...\}$, we can detect a NAT device's translation algorithm based on the correlation of sequences [13, 4.4 Advantages of the New Method]. If a client is behind random symmetric NAT, a random port inside the port range is predicted. The traversal rate is improved by opening a large number of opened ports from the client mapped through random symmetric NAT.

Yao, Hwang, and Yeh [14] develop a mathematical model to enhance the port predictability of NAT and increase the success rate of NAT traversal. Besides linear prediction targeting at progressing symmetric NAT, they provide a model for nonlinear port mapping for two-level linear port mapping based on n-th order-jump functions. However, our measurement study indicated linear progressing or random symmetric NAT only.

Huang et al. [4] present a more complex prediction method for progressing symmetric NAT translated ports in a recent work from 2019. Based on port allocation traffic and the elapsed time, the so-called 'network traffic rate' λ is calculated for each peer to predict ports by means of either the Expected Value Method (EVM) or the Poisson Sampling Method (PSM). More precisely, the 'network traffic' corresponds to the number of new port allocations over time. Klinec and Matyáš [5] proposed to model the internal network connected to NAT with respect to newly created connections, i.e. a new port allocation, as a Poisson process with rate λ. While EVM predicts linear

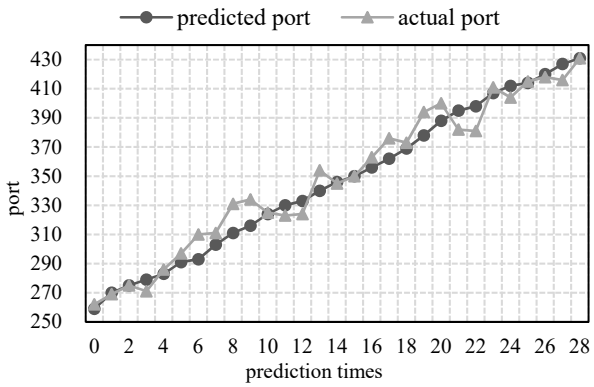

Figure 1: Comparison of predicted ports using the Poisson Sampling Method (PSM) and actual ascending symmetric NAT translated ports [4]. Copyrights with the original author, licensed under CC BY-NC 4.0.

port successions, PSM is based on a probability distribution to allow for minor deviations. Both methods predict a list of remote ports based on a fixed time interval. For each element of the list, a traversal packet is sent to the other peer targeting the predicted port at the respective index. Figure 1 shows a sequence of predicted ports using PSM and the actual ports that are allocated in a constantly changing range around the predicted ports. If ports are predicted correctly for both peers at the same iteration, a communication is established. Both described methods have strong results in experimental environments, but their traversal rates can suffer in real world environments from high latencies, inconsistent NAT behavior or rapidly changing traffic rates. This paper extends EVM and PSM based on the doubled exchange delay for progressing symmetric NAT and a new heuristic approach for random symmetric NAT. All methods are optimized for real-world tests inside the German mobile network.

III. Structure of the Proposed Method

For symmetric NAT traversal, we propose a procedure of matchmaking, analysis and traversal (MAT) as depicted in Figure 2, which features peers with two roles (Host and Client) as well as a server for exchanging information. At the beginning of the matchmaking phase, each peer has to check its NAT type. Each peer sends two requests from the same private address and port to different listening ports of the server. The server determines the type of NAT based on the translated remote ports of the NAT device and forwards the result to the respective peer. In the last step of the matchmaking phase, the peers inform the server about the role they want to play, either Client or Host. Hosts can be seen by all peers, whereas Clients can connect to all available Hosts.

If the identified NAT situation includes at least one progressing symmetric peer, the analysis phase starts with measuring the time for exchanging peer information relayed by the server. We expect the elapsed time to approximate an actual exchange of network information between peers. This delay is later used

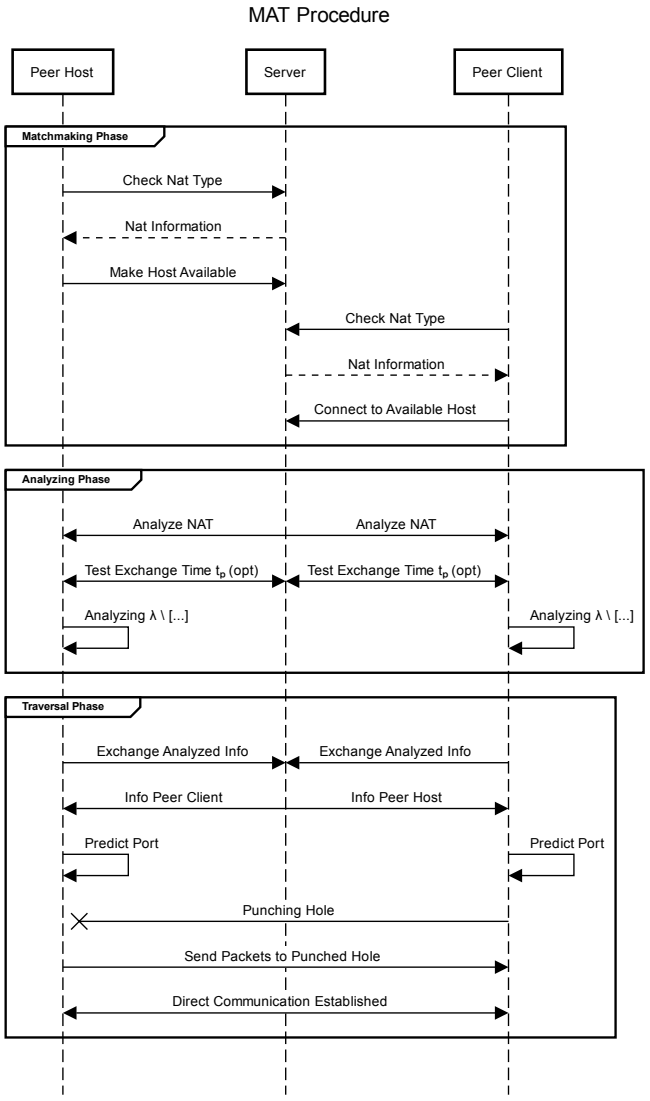

Figure 2: The MAT procedure to punch through a symmetric NAT. MAT stands for matchmaking, analyzing and traversing.

to predict progressing ports. Next, each peer analyzes its NAT. When a peer is behind progressing symmetric NAT, its current port allocation traffic is analyzed. In the random symmetric NAT case, samples of opened remote ports are collected. Both methods are further discussed in the upcoming subsections.

In the traversal phase, the analyzed data from each peer is exchanged with the help of the server. Based on the provided information, a port subsequently allocated by the other peer is predicted, which is used as target port in the hole-punching process. If both peers have predicted an actually opened port by the NAT devices, the communication is established. The specific implementation of the port prediction process is discussed in the following sections.

A. Traffic Analysis of Progressing Symmetric NAT Devices

For analyzing the current port allocation traffic, multiple samples of NAT translated ports are collected. The traffic rate is calculated based on the port differences from the samples and the time consumption. It is measured in opened ports per millisecond. As model for the internal network connected to NAT with respect to newly created connections, Klinec and Matyáš [5] suggests a Poisson process with network traffic rate λ. Unfortunately, port allocation in progressing symmetric NAT is not always performed in a progressing order. These leaps in port numbers take place, when ports were already allocated by a different program before. In order to overcome this problem, the traffic rate calculation uses a high number of remote ports, collected with a small Δt. If between two sequential port samples the delta port number is greater than a maximum value, just a single port is added to the sum of elapsed port numbers. Otherwise the actual delta port number is added to the sum. All ports opened by the calculation are subtracted from the sum of traffic ports to get the correct idle traffic. The described approach is summarized in Algorithm 1.

Algorithm 1 Traffic Calculation

```
procedure GETTRAFFIC(samples, time, maxDelta)
    sumPorts ← 0
    for port, nextPort in samples do
        deltaPorts ← delta(port, nextPort)
        if deltaPorts < maxDelta then
            sumPorts ← sumPorts + deltaPorts
        else
            sumPorts ← sumPorts + 1
    return (sumPorts - length(samples)) / time
```

IV. Port Prediction in Edge-Case Situations

In this section, we detail the port prediction techniques for progressing symmetric and random symmetric NAT.

A. Traversal Method for Progressing Symmetric NAT

The port prediction for a peer connected with progressing symmetric NAT is based on the last known remote port, the earlier discussed port allocation traffic rate and the predicted time for passing messages to the other peer. As soon as the information has reached the other peer, the doubled exchange delay method (DEVM) is used to make a port prediction, see next section. Based on the resulting port, the hole-punching process is executed. In order to improve the chance of a successful traversal, many traversal packets are sent without any extra delay between the single socket operations.

B. Doubled Exchange Delay Method

In the first step of the prediction process, the number of elapsed ports $\Delta PORT_t$, that were opened due to the general network traffic λ in the communication period, is calculated.

$$\Delta PORT_t = round(t_p \cdot \lambda) \tag{1}$$

Equation (1) shows the calculation, which uses the predicted delay t_p for exchanging peer information and the earlier monitored network traffic rate λ.

In the second step, additional ports are added to reduce the risk of predicting a remote port, that is already allocated by a different program. $\Delta PORT_a$ is calculated using the Expected Value Method (EVM, Equation 2) or the Poisson Sampling Method (PSM [4]). In both methods the ith predicted port from the calculation of the sampling list is chosen as $\Delta PORT_a$. EVM and PSM are considered in the evaluation results of our procedure.

$$\Delta PORT_a = i + i \cdot \lambda \cdot \Delta t \tag{2}$$

Equation (2) uses a time interval Δt, the current network traffic λ and the target iteration i. Dependent on the length of the time interval Δt, more or less ports are added to the predicted port. It can be seen as the temporal distance between opening two consecutive ports in a progressing NAT situation (e.g. 10 milliseconds). With the help of the predicted latency for exchanging peer information t_p and the time interval Δt, the target iteration i can be calculated shown in Equation (3).

$$i = round(t_p/\Delta t) \tag{3}$$

Finally, Equation (4) calculates the final predicted port ($predPORT$) using the exchanged remote port of the other peer ($natPORT$), the elapsed ports due to network traffic ($\Delta PORT_t$) and the additional ports ($\Delta PORT_a$). The predicted port calculation is based on the doubled exchange delay, because $\Delta PORT_t$ and $\Delta PORT_a$ are both depending on this value. The other peer connected via progressing symmetric NAT has to open a reasonable number of ports to cover the $predPORT$ value.

$$predPORT = natPORT + \Delta PORT_t + \Delta PORT_a \tag{4}$$

The predicted target address and port consists of the remote address of the other peer and the $predPORT$ value. While Huang et al. [4] is predicting pairs of ports in multiple time steps, we are predicting a single port, which overestimates the actual allocated port based on t_p, Δt and λ. The progressing symmetric peer is opening a reasonable number of ports at once, covering the predicted port to establish a communication.

C. Traversal Method for Random Symmetric NAT

The prediction for random symmetric NAT is based on the remote address and a random sample port from the collected port list in the analysis phase of the peer. The target port for the traversal packets is randomly chosen between the sample port + 1 and the sample port + 2. The reason for this approach is discussed in Section VI. After the prediction phase, the random symmetric peer is opening a large amount of ports by sending traversal packets to increase the probability to open the predicted port. Instead of sending all packets at once, chunks of traversal packets from the random symmetric peer are sent in multiple iterations. Each iteration is followed by a delay to avoid triggering the flooding protection of the other peer.

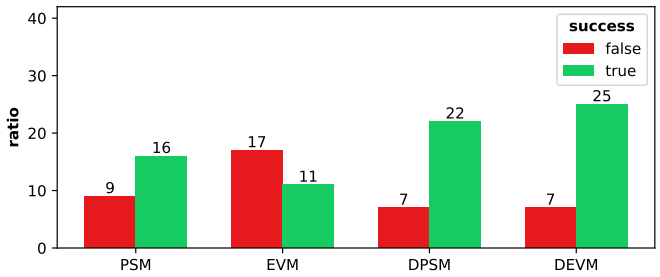

Figure 3: Traversal result of two peers behind progressing symmetric NAT using PSM and EVM [4] in combination with the new doubled exchanged delay method (DPSM, DEVM).

V. Methodology

We conducted tests using the User Datagram Protocol (UDP) in two European cities with Internet access via 4G of three mobile operators (Telefonica, Vodafone, Telekom). Table I shows the discovered NAT types of the operators.

Provider	NAT Type Discovered
Telekom	Ascending Symmetric
Telefonica	Random Symmetric
Vodafone	Random Symmetric

Table I: Evaluated NAT types for cellular network providers.

Only the tested mobile network provider Telekom Deutschland GmbH is using progressing symmetric NAT. In the performed tests, clients were located in different cities with a distance of around 150 km to prevent the usage of the same remote address. All other tests were located in just one city, using a different mobile-network provider for each client (Telefonica, Vodafone, Telekom). The tests were executed on HP Omen 17-w106ng and Lenovo ThinkPad T480s Windows laptops connected to the internet by mobile hotspots created by means of iPhone 7 Plus, iPhone 11 and Sony Xperia XZ1 devices or by home routers. The home router was behind port restricted Cone NAT. The server is located in a metropolitan area about 100 km away. The implementation for server and clients was written in Python using the standard libraries. The following constants were used: The temporal distance between the server messages in the NAT identification process is a configuration parameter of our procedure and set to 10 ms. The network traffic rate for progressing symmetric NAT is calculated from 100 requests to the server with a temporal distance of 8 ms. In the traversal process for progressing symmetric and random symmetric NAT, 150 and 3,000 packets are sent. In case of random symmetric NAT, the packets are sent in 17 iterations, each comprised of 180 traversal packets followed by a delay of 500 ms. Peers behind non-symmetric NAT are sending 30 traversal packets, using the same port number. A TTL value of 8 is used for dropping packets in the hole-punching process.

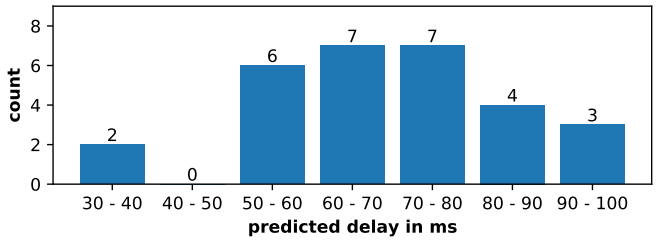

Figure 4: Average latencies for exchanging peer information in the executed tests for traversing progressing symmetric NAT.

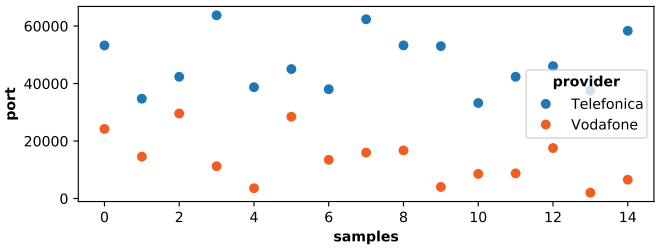

Figure 5: Random symmetric NAT allocated ports.

VI. Experiments

Four different methods for traversing peers behind progressing symmetric NAT are evaluated. The first two implementations use the Expected Value Method (EVM) and the Poisson Sampling Method (PSM) and their settings from Huang et al. [4]. Figure 3 shows the successful traversal rates of PSM and EVM of 64% and 39%, respectively. Their extensions by the double exchange delay, i.e. DPSM and DEVM, reach about 76% and 78%, respectively, only predicting a single port. One reason for the lower success rates of regular EVM and PSM are high latencies when exchanging peer information. Figure 4 shows the average predicted exchange delay for this specific test case is between 60 ms to 80 ms. A long time between measuring the current traffic rate and sending the first traversal packet can have a negative effect on the plausibility of the computed traffic rate. In our tests the port allocation traffic can change dramatically in small time steps. DEVM and DPSM are more robust against this problem, because the single predicted port is less time dependent. As long as the targeted port is higher or equal to the actual open port in the traversal phase, a communication link can be established.

In cases when one peer is behind a Cone type of NAT and the other peer is behind progressing symmetric NAT, DEVM has a traversal rate of approximately 93% in the performed tests. The traversal rate is increased to earlier tests, because we already know the specific port mapping of the full/restricted peer.

In the following, traversal rates between random symmetric and a Cone type of NAT are evaluated. Figure 5 shows a small amount of consecutively, randomly allocated ports. Based on a large sampling set, it is obvious that the tested providers allocate only within half of the available port range. This behavior can improve the success rate of many traversal algorithms, yet, the delta values between consecutive ports cover a large part of the port range. This makes it difficult to predict the next allocated port based on the currently monitored one. One method to attempt this predicts 30 random ports inside the port range (labeled 'TelR', 'VodR'), another one predicts ports in the range of +/-15 around the smallest port monitored during the analysis phase (labeled 'TelM', 'VodM'). The random symmetric peer opens 3,000 ports and the full/restricted peer is opening 30 ports using the same private address and port. In Figure 6 (a), both methods show similar traversal rates with about 74% and 73% in the Telefonica and 77% and 83% in the Vodafone mobile network. Random port allocation makes it much harder to achieve reasonable traversal rates using a small amount of open ports.

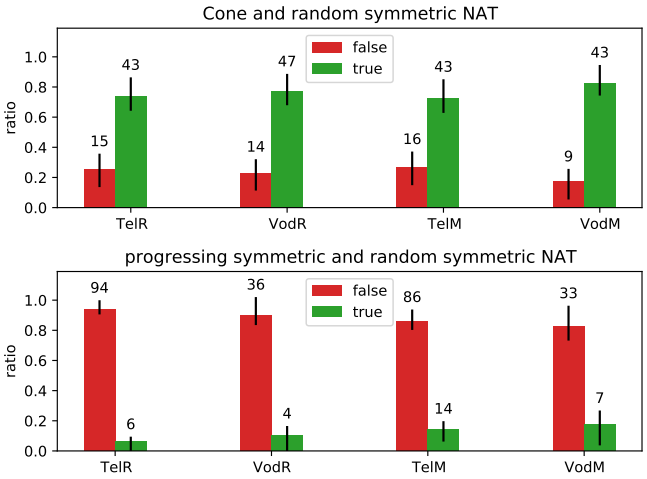

Figure 6: Ratio of (a) traversing full/restricted Cone and random symmetric NAT (upper figure) and (b) random symmetric and progressing symmetric NAT (lower figure) in the mobile networks of Telefonica and Vodafone. Absolute numbers are given above the bars.

Figure 7 shows the sum of the successful traversal attempts for the different delta ports. Both mobile providers share a similar result. The distribution of successful traversals for the range around the monitored port peeks in the close positive range (for Telefonica at delta port + 1, Vodafone at delta port + 2). The result shows that port allocation in random symmetric NAT is not evenly distributed which can be exploited to improve traversal rates. In order to predict a single target port for random symmetric NAT, a random port is chosen between the analyzed smallest port + 1 and + 2. In the following test, the second peer is behind progressing symmetric NAT and traversed with DEVM. Figure 6 (b) shows the traversal rate of 6% and 10% when predicting a random port, and a success rate of 14% and 17.5% with the proposed method. The different success rates between the providers can be explained by the different port allocation algorithms and traffic rates. The proposed method can be applied to all monitored ports in the analysis phase and is not just limited to the smallest one.

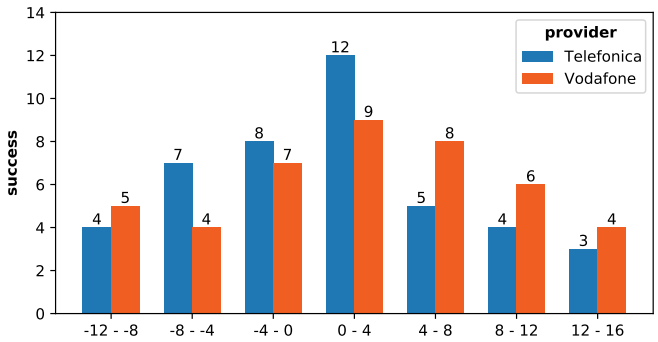

Figure 7: Frequently translated delta ports of random symmetric NAT based on the smallest monitored port are shown. The success count of a delta port was increased for each successful traversal attempt with this delta port. The maximum is in the close positive range around the analyzed port.

When applying the proposed technique for traversing random symmetric NAT, the success rate is not significantly improved to prediction methods with random ports.

VII. Conclusion

This paper presents an extension to established direct communication protocols for symmetric NAT edge-cases in real-world environments. The proposed hole-punching approach predicts progressing symmetric NAT allocated ports based on timing and traffic. Port predictions for random symmetric NAT are based on a large amount of traversal packets and a heuristic. In field tests, the traversal of two random symmetric peers had insufficient traversal success rates. However, there is a significant improvement for traversing progressing symmetric and random symmetric NAT, with a success rate between 14 and 17.5% compared to approaches using random ports with 6 and 10%. In all other edge-cases, the traversal success rate is at least 73%. All presented traversal methods have increased success rates in the performed tests, compared to traditional approaches. This paper shows the applicability of establishing direct communications in symmetric NAT scenarios in real-world situations. The traversal method is also applicable for TCP, using an adjusted TCP hole-punching method suitable for mobile networks [11]. In our tests, both protocols used the same port range. The method presented here should be tested in more symmetric NAT environments to prove their general applicability. Further research is necessary for predicting random symmetric NAT allocated ports, in order to accomplish strong traversal results in all cases. Neural networks could be used to perform the prediction of upcoming translated ports from random symmetric NAT devices.

References

[1] F. Audet and C. Jennings. *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*. BCP 127. RFC Editor, Jan. 2007.

[2] Kjeld Borch Egevang and Paul Francis. *The IP Network Address Translator (NAT)*. RFC 1631. RFC Editor, May 1994.

[3] Arijit Ganguly et al. “Improving peer connectivity in wide-area overlays of virtual workstations”. In: *Cluster Computing* 12.2 (2009).

[4] Jiayu Huang et al. “Study of A Novel Predictable Poisson Method for Traversing Symmetric NAT”. In: *Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019)*. Atlantis Press, 2019/07.

[5] Dušan Klinec and Vashek Matyáš. “Traversing symmetric NAT with predictable port allocation”. In: *Proceedings of the 7th International Conference on Security of Information and Networks*. 2014.

[6] R. Mahy, P. Matthews, and J. Rosenberg. *Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)*. RFC 5766. RFC Editor, Apr. 2010.

[7] Andreas Müller, Andreas Klenk, and Georg Carle. “On the applicability of knowledge based nat-traversal for home networks”. In: *International Conference on Research in Networking*. Springer. 2008.

[8] J. Rosenberg. *Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols*. RFC 5245. RFC Editor, Apr. 2010.

[9] J. Rosenberg et al. *Session Traversal Utilities for NAT (STUN)*. RFC 5389. RFC Editor, Oct. 2008.

[10] Branislav Sredojev, Dragan Samardzija, and Dragan Posarac. “WebRTC technology overview and signaling solution design and implementation”. In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Opatija, Croatia: IEEE, May 2015, pp. 1006–1009.

[11] Satish Narayana Srirama and Mohan Liyanage. “TCP Hole Punching Approach to Address Devices in Mobile Networks”. In: *2014 International Conference on Future Internet of Things and Cloud*. Barcelona, Spain: IEEE, Aug. 2014.

[12] Yong Wang, Zhao Lu, and Junzhong Gu. “Research on symmetric NAT traversal in P2P applications”. In: *2006 International Multi-Conference on Computing in the Global Information Technology-(ICCGI’06)*. IEEE. 2006.

[13] Yuan Wei et al. “A New Method for Symmetric NAT Traversal in UDP and TCP”. In: (Jan. 2008).

[14] Bing-Jhih Yao, Shaw-Hwa Hwang, and Cheng-Yu Yeh. “Mathematical model of network address translation port mapping”. In: *AASRI Procedia* 8 (2014).

[15] L. Zhang. “A retrospective view of network address translation”. In: *IEEE Network* 22.5 (2008).