Использование NAT в межсетевых экранах

классификация, принципы функционирования, трудности для сетевых приложений

Михалец Мартин, гр. 5130201/20101[1]

2026-03-12

ПОЛИТЕХ

Санкт-Петербургский политехнический университет Петра Великого

[1] martin@michalec.dev

Введение

Что такое NAT и зачем он нужен:

- NAT (*Network Address Translation*) – «преобразование сетевых адресов»
- Позволяет множеству внутренних устройств использовать один внешний публичный адрес
- Используется повсеместно, т.к. IPv4-адресов недостаточно
- Простейшая форма защитного барьера

Другие названия:

- IP Masquerading
- Network Masquerading
- Native Address Translation

Цель и задачи

Цель:

Исследовать особенности использования технологии NAT в межсетевых экранах, раскрыть её влияние на работу сетевых приложений, а также проанализировать методы обхода ограничений NAT.

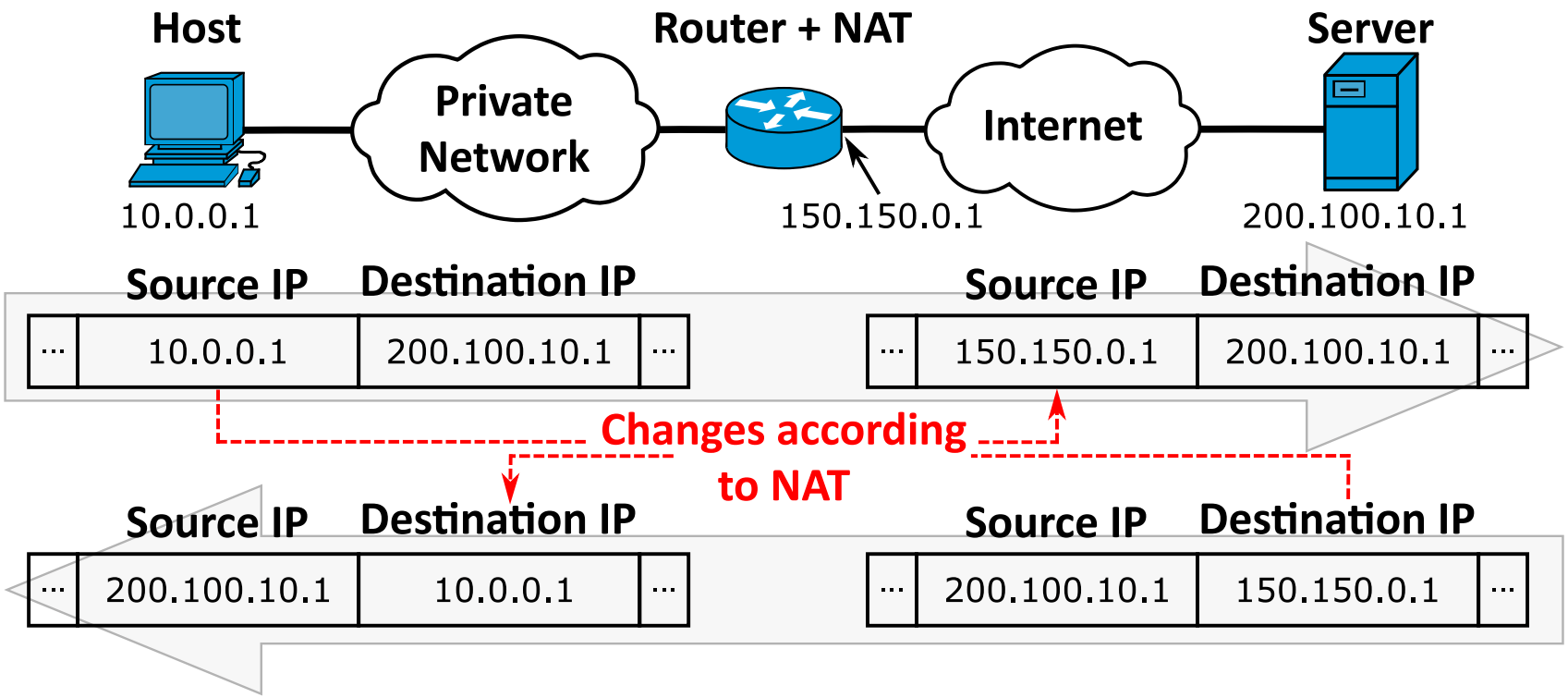
Задачи:

1. Изучить сущность и назначение NAT
2. Классифицировать основные виды NAT
3. Проанализировать влияние NAT на сетевые приложения
4. Рассмотреть специфику построения P2P-соединений в сетях с NAT
5. Описать методы NAT-traversal

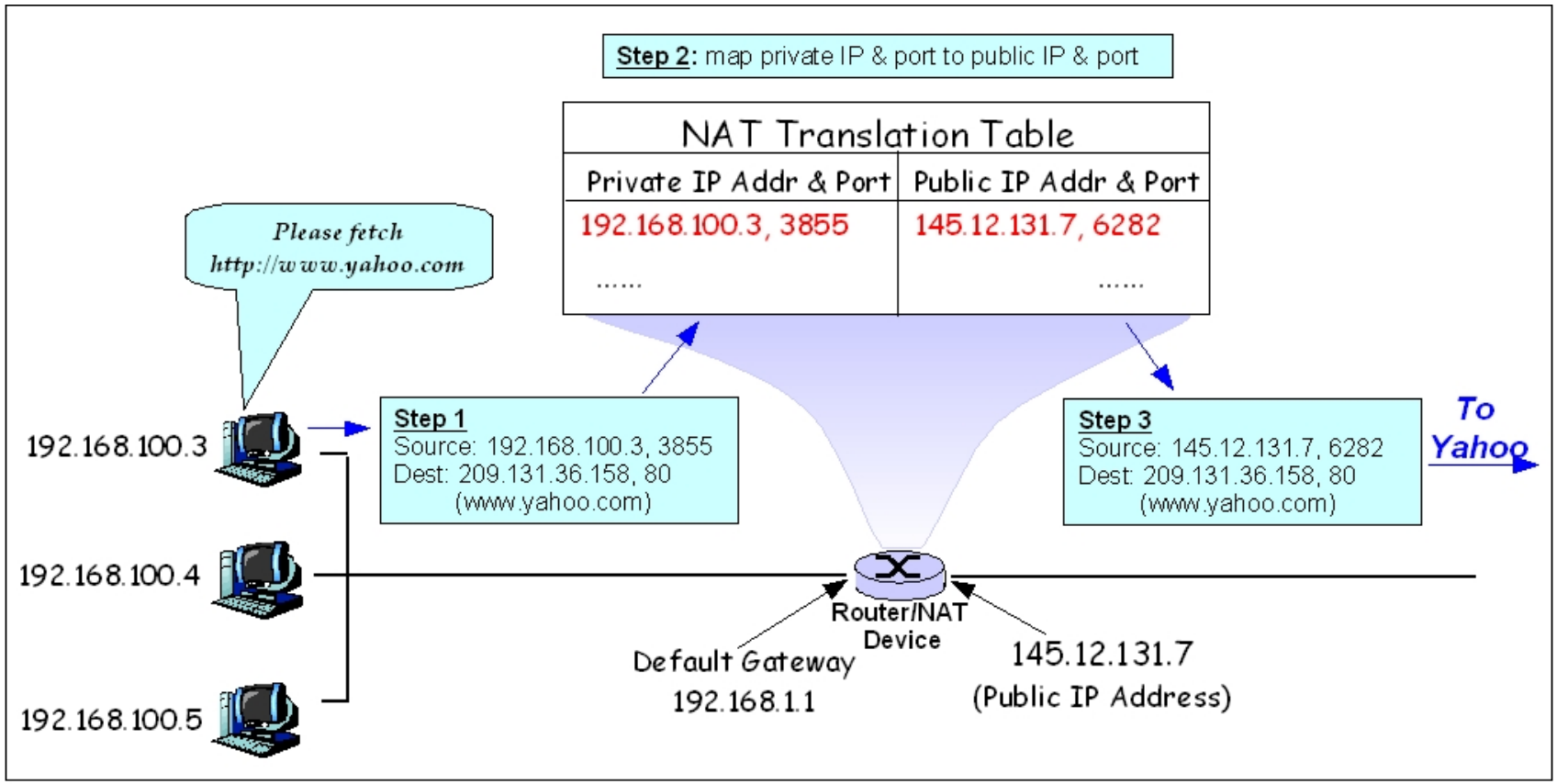
Классификация по концепции NAT

- Статический NAT – отображение незарегистрированного IP-адреса на зарегистрированный IP-адрес на основании один к одному.
- Динамический NAT – отображает незарегистрированный IP-адрес на зарегистрированный адрес из группы зарегистрированных IP-адресов.
- Перегруженный NAT (NAPT, NAT Overload, PAT, маскарадинг) – форма динамического NAT, который отображает несколько незарегистрированных адресов в единственный зарегистрированный IP-адрес, используя различные порты.

Network Address Translation



Network Address Translation



NAT в межсетевых экранах и безопасность

NAT сам по себе повышает безопасность:

- Защищает внутренние устройства от неожиданных входящих соединений
- Скрывает внутреннюю топологию сети

Но NAT — не полноценная безопасность:

- Он не блокирует вредоносные исходящие соединения
- Безопасность всегда дополняется фильтрацией пакетов и политиками FW

Классификация по типу NAT[3]

Таблица 1: С точки срения функционирования

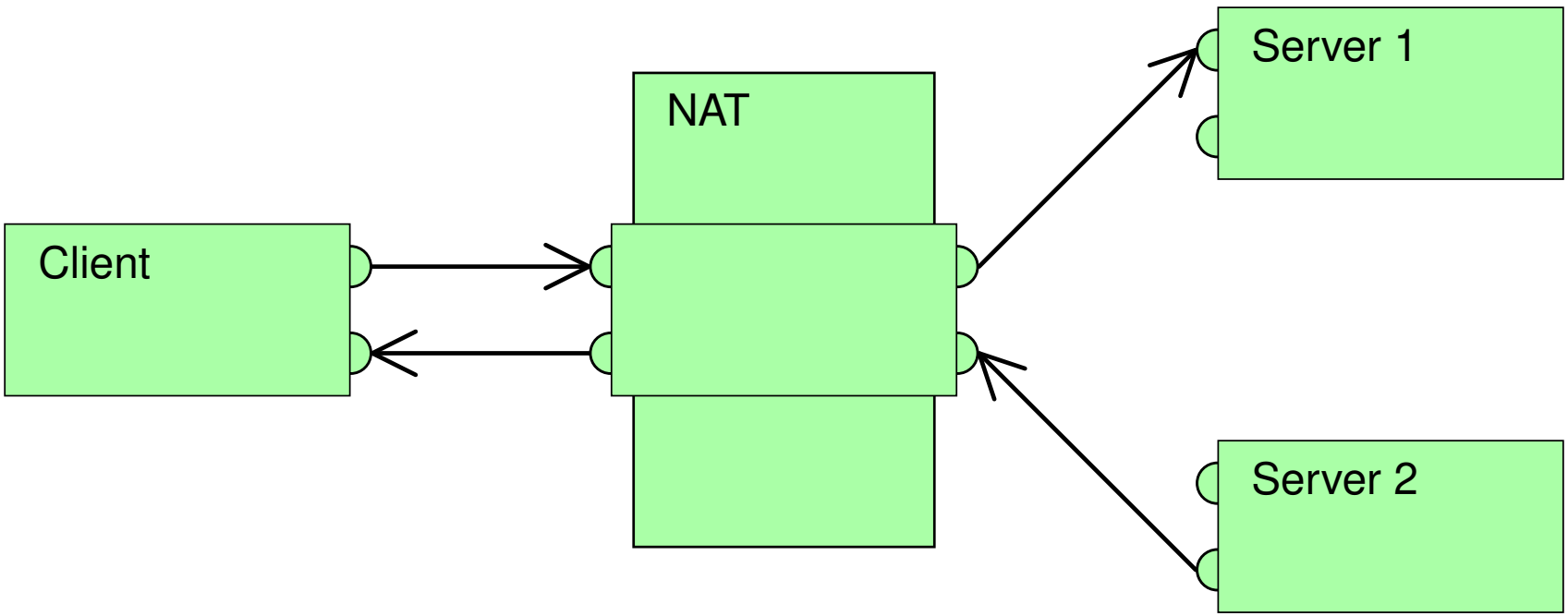
Filtering \ Mapping	Endpoint-Independent	Endpoint-Dependent
Endpoint-Independent	Full Cone NAT	N/A[2]
Endpoint-Dependent (dest. IP only)	Restricted Cone NAT	N/A[2]
Endpoint-Dependent (dest. IP+port)	Port-Restricted Cone NAT	Symmetric NAT

[2] теоретически могут существовать, но не встречаются

[3] более подробно в RFC 4787, 5382, 5508

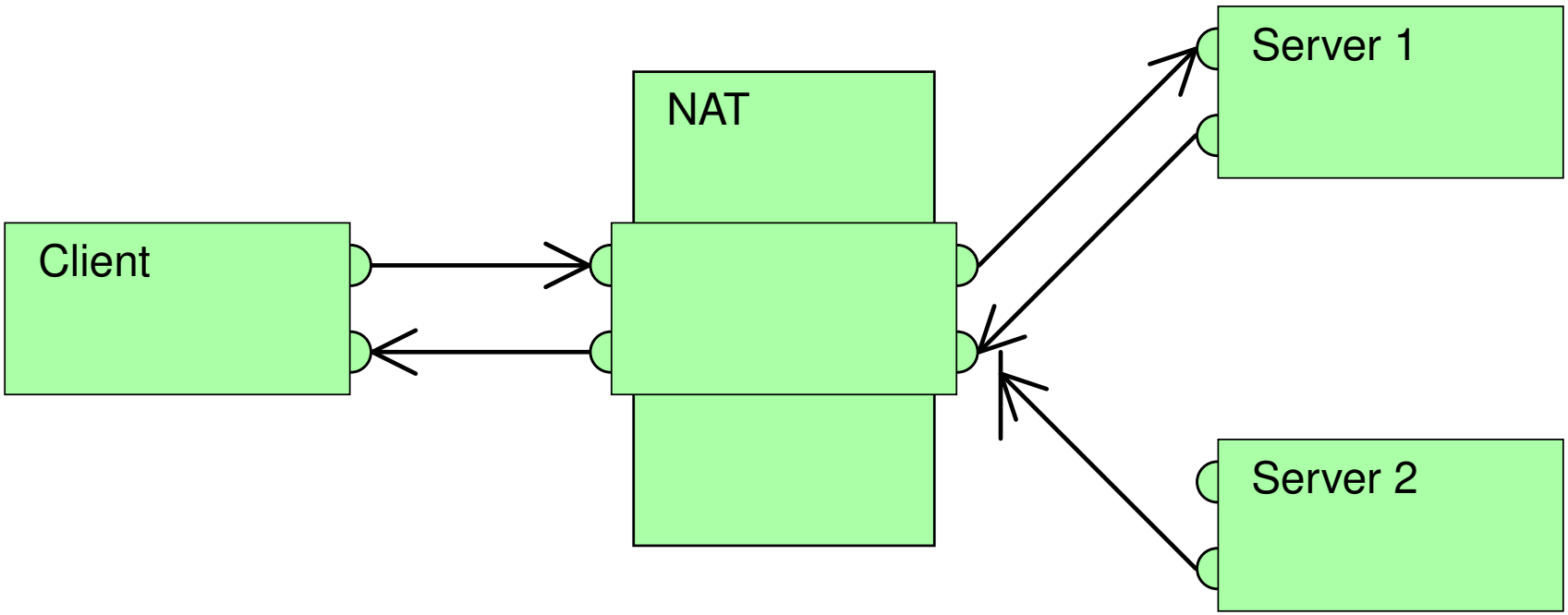
Full Cone NAT

Однозначная (взаимная) трансляция между парами «внутренний сокет» и «публичный сокет». Любой внешний хост может инициировать соединение с внутренним хостом (если это разрешено в правилах межсетевого экрана).



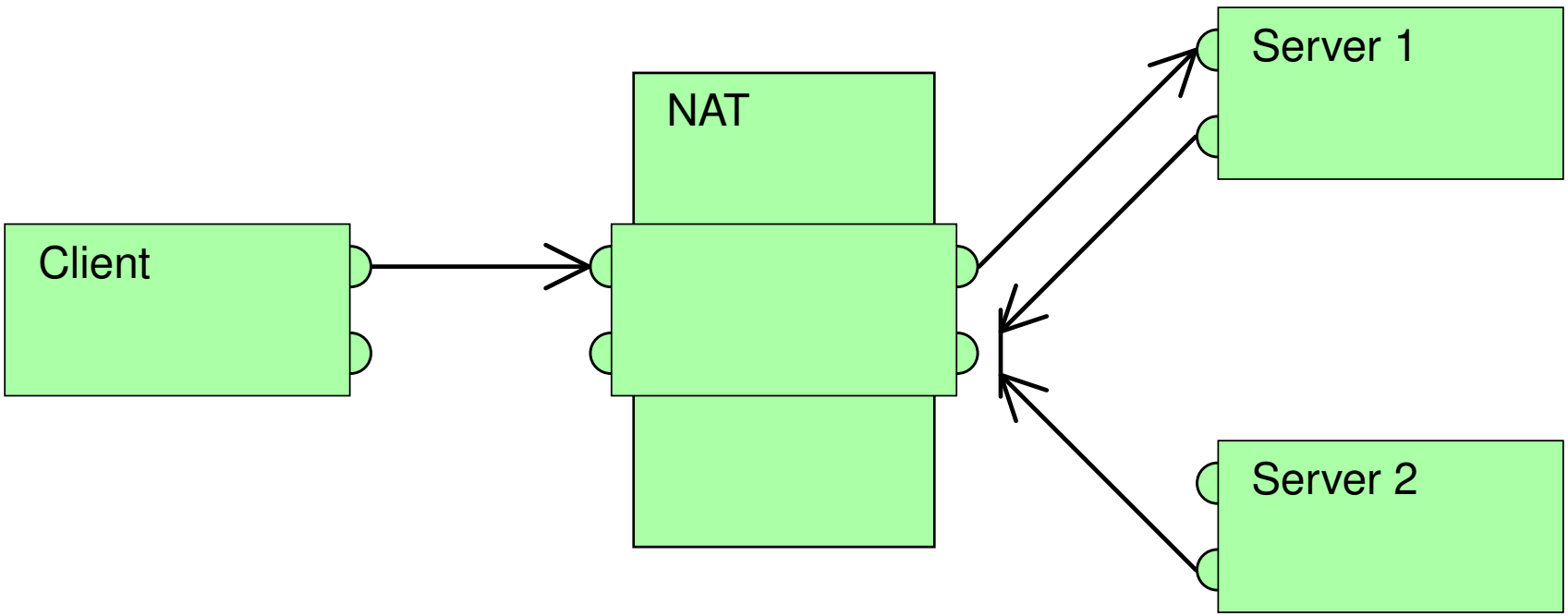
Restricted Cone NAT

Постоянная трансляция между «внутренний сокет» и «публичный сокет». Любое соединение, инициированное с внутреннего адреса, позволяет в дальнейшем получать ему пакеты с любого порта того публичного хоста, к которому он отправлял пакет(ы) ранее.



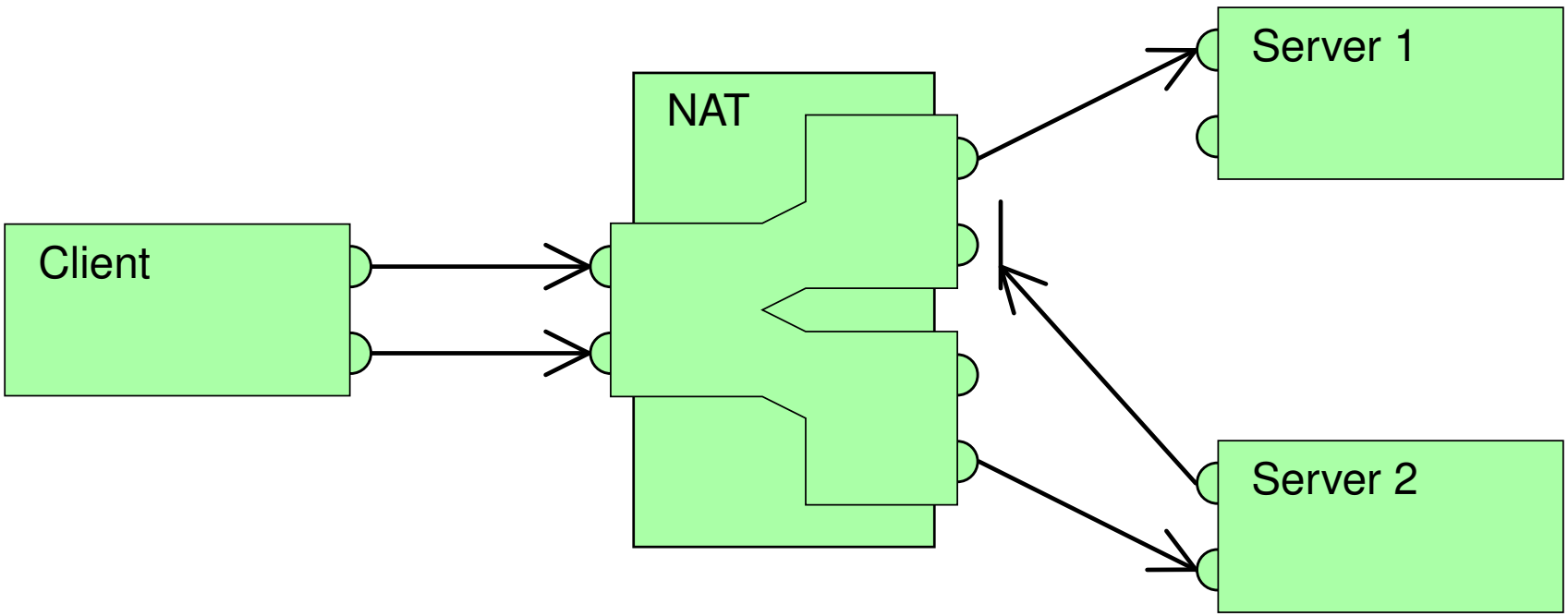
Port Restricted Cone NAT

Transcripce mezi «vnitřní sokét» a «veřejný sokét», při které příchozí pakety procházejí na vnitřní hostitel pouze z jednoho portu veřejného hostitele – toho, na který vnitřní hostitel již poslal paket.

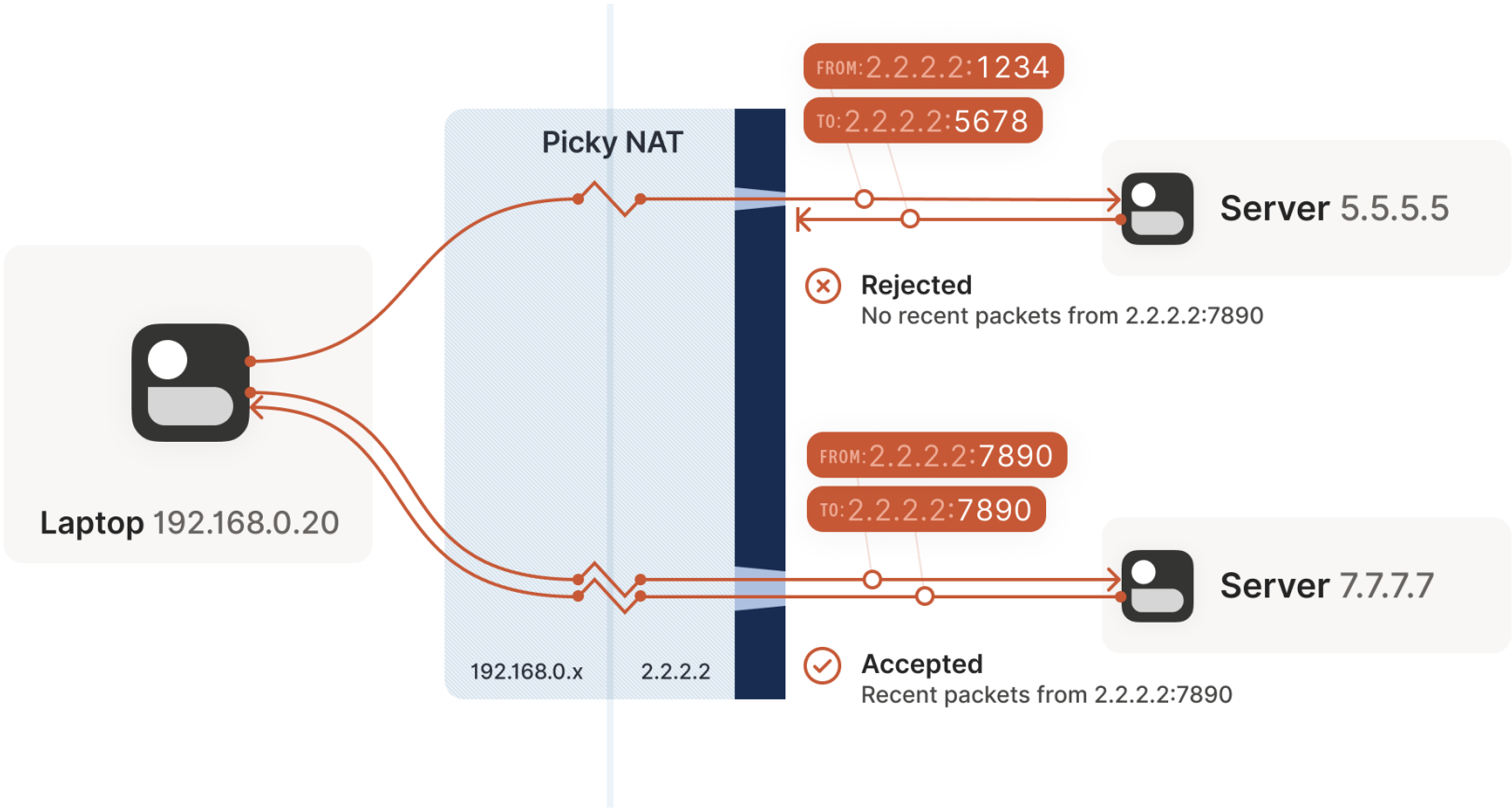


Symmetric NAT

Трансляция, при которой каждое соединение, инициируемое парой «внутренний сокет», преобразуется в свободную уникальную случайно выбранную пару «публичный сокет». При этом инициация соединения из публичной сети невозможна.



Symmetric NAT



Другие классификации NAT

- SNAT (Source NAT) – меняет сокет отправления
- DNAT (Destination NAT) – меняет сокет назначения
- CGNAT (Carrier-grade NAT) – многослойный
- NAT44 – IPv4 $\leftrightarrow$ IPv4
- NAT66 – IPv6 $\leftrightarrow$ IPv6
- NAT46 – IPv4 $\leftrightarrow$ IPv6
- NAT64 – IPv6 $\leftrightarrow$ IPv4

Трудности для сетевых приложений

Проблема:

NAT по умолчанию не позволяет входящие соединения, если не было предварительного исходящего запроса.

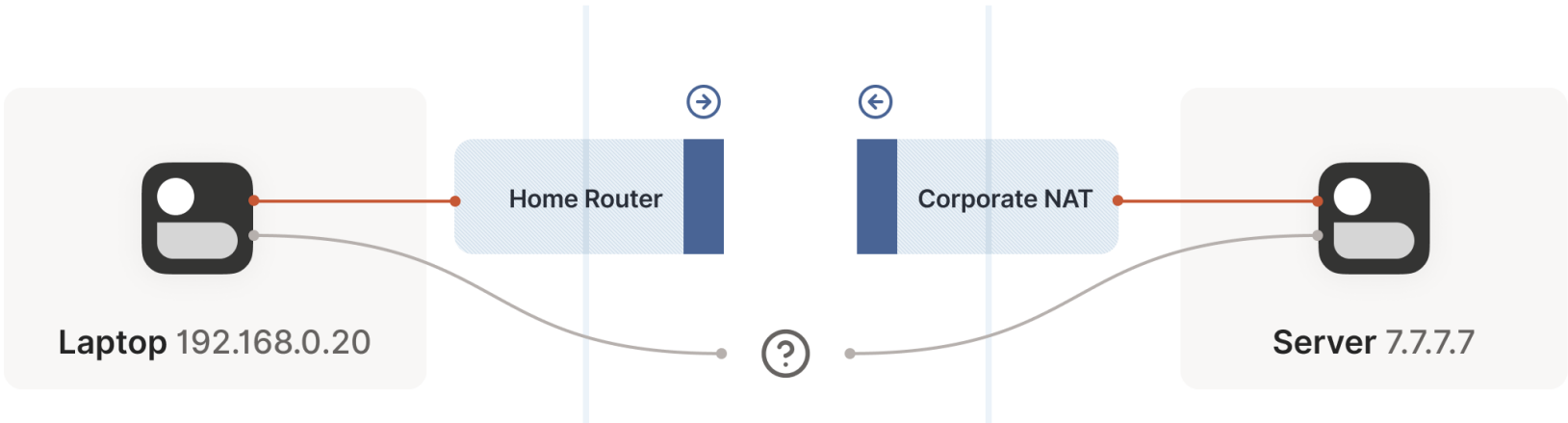
Последствие:

Приложения, которые предполагают, что любой узел может выступать и как клиент, и как сервер, оказываются сломаны в сети с NAT.

NAT Traversal

Что такое NAT Traversal:

Набор методов, позволяющих двум пирами, находящимся за NAT, найти друг друга и установить канал связи несмотря на межсетевые экраны



NAT Traversal

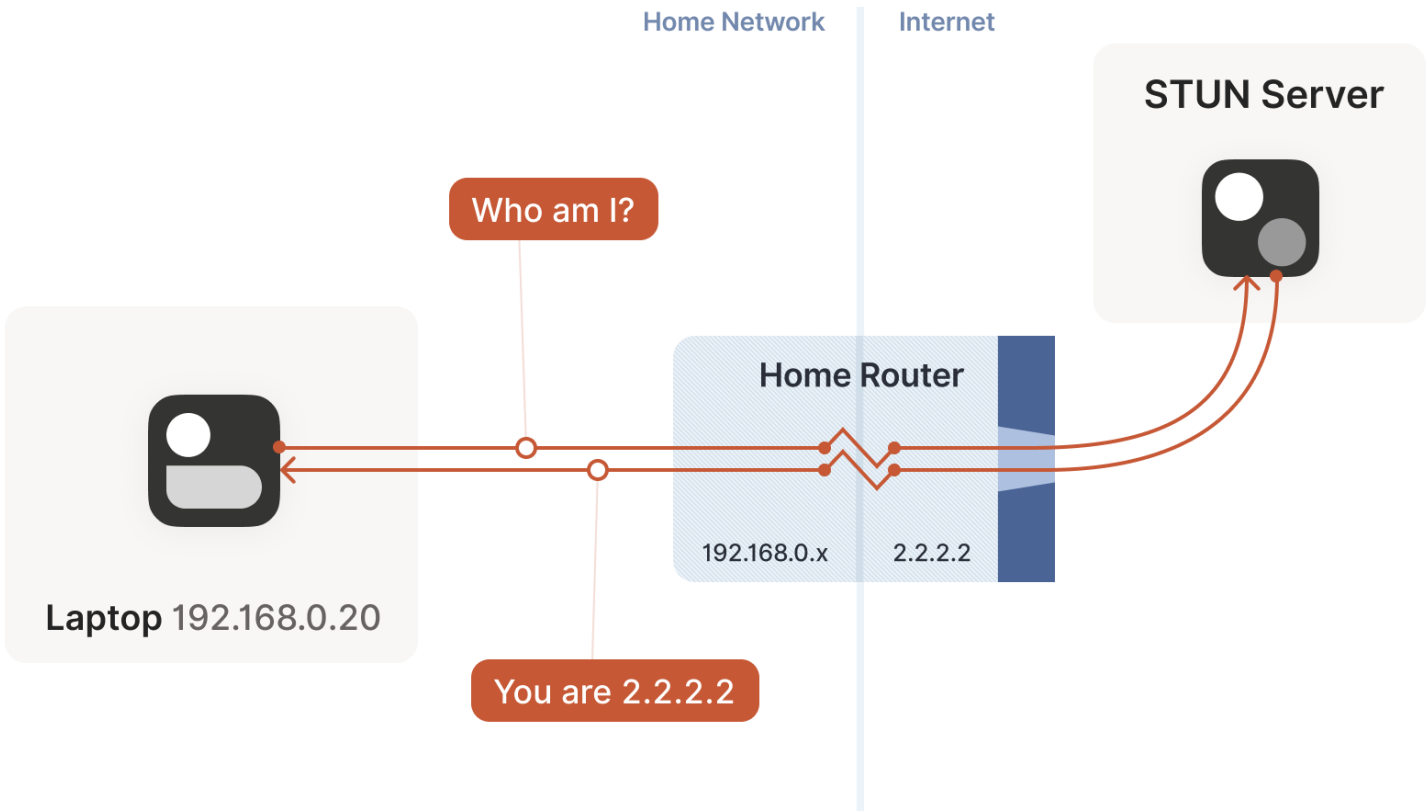
Основные техники:

- Side Channel – для координации
- STUN[4] – обнаружение внешнего IP:порт и типа NAT
- UDP/TCP Hole Punching[5] – «пробивание дыр» в NAT-таблицах для создания двунаправленного канала
- TURN[6] – трафик передаётся через сервер-ретранслятор
- ICE[7] – объединяет STUN + TURN и выбирает оптимальный маршрут подключения

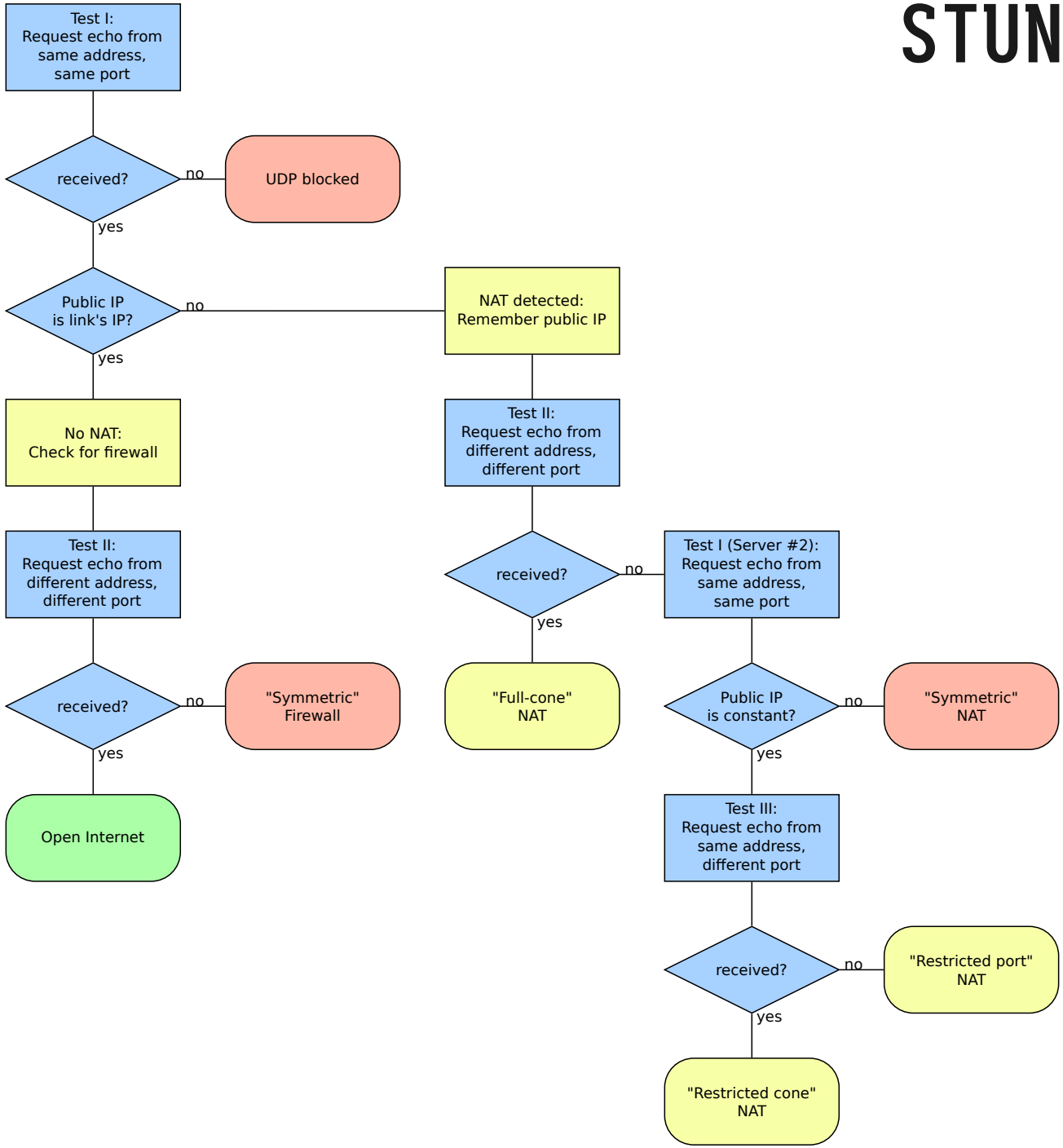
---

[4] RFC 8489, Session Traversal Utilities for NAT
[5] RFC 5128
[6] RFC 8656, Traversal Using Relays around NAT
[7] RFC 8445, Interactive Connectivity Establishment

Session Traversal Utilities for NAT



STUN


Test I:
Request echo from
same address,
same port
received?
no
UDP blocked
yes
Public IP
is link's IP?
no
NAT detected:
Remember public IP
yes
No NAT:
Check for firewall
Test II:
Request echo from
different address,
different port
Test II:
Request echo from
different address,
different port
received?
no
"Symmetric"
Firewall
yes
Open Internet
received?
no
Test I (Server #2):
Request echo from
same address,
same port
yes
"Full-cone"
NAT
Public IP
is constant?
no
"Symmetric"
NAT
yes
Test III:
Request echo from
same address,
different port
received?
no
"Restricted port"
NAT
yes
"Restricted cone"
NAT

UDP/TCP Hole Punching

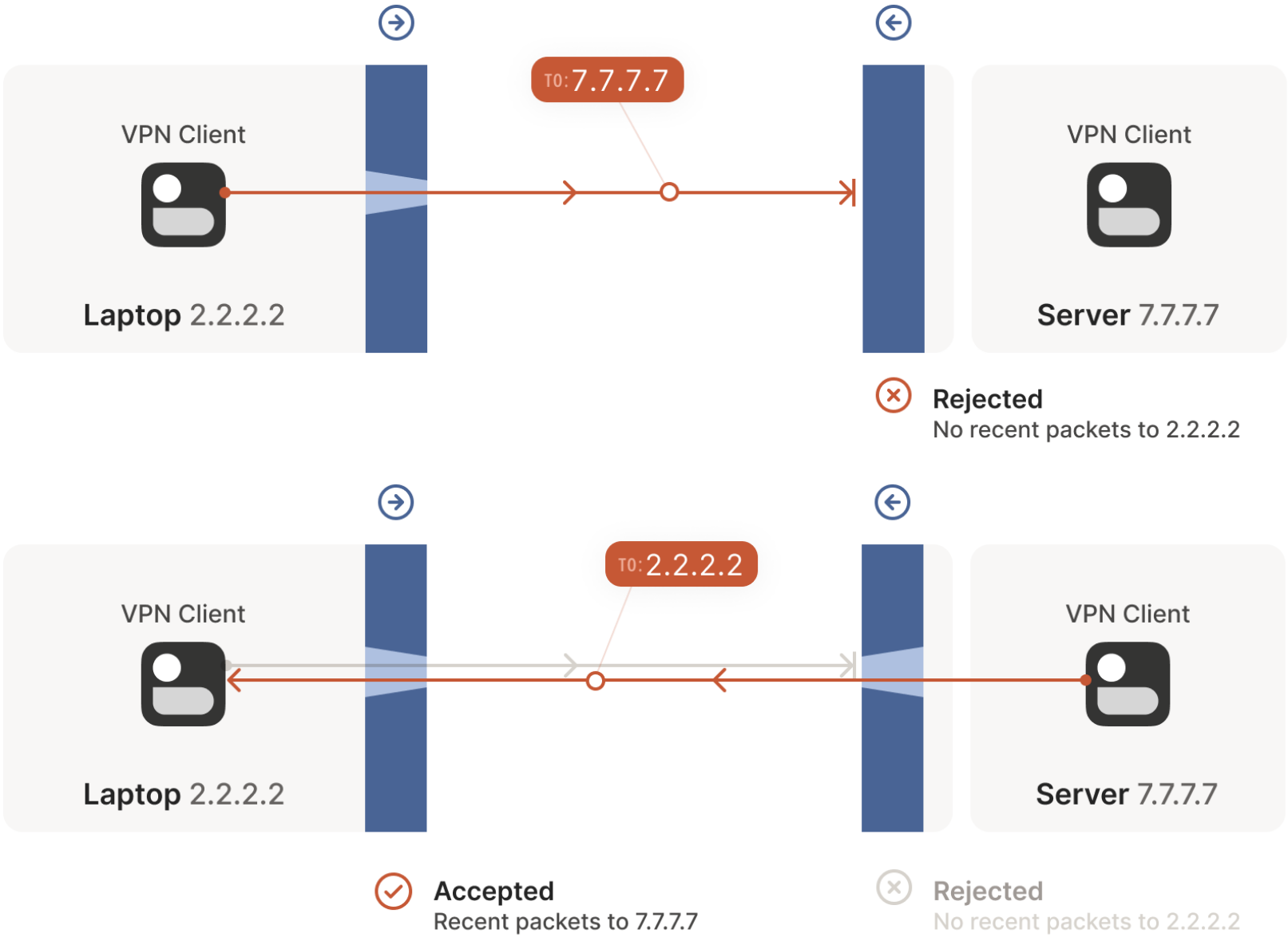
Идея:

Каждый пир отправляет исходящие пакеты одновременно так, что NAT создаёт табличные записи, принимая последующие входящие пакеты как ответы на исходящие.

Проблемы с TCP:

- NAT может отправлять RST в ответ на несуществующие TCP SYN
- Сложность с предсказанием портов для двустороннего открытия TCP-соединения (simultaneous open)
- ...

UDP/TCP Hole Punching



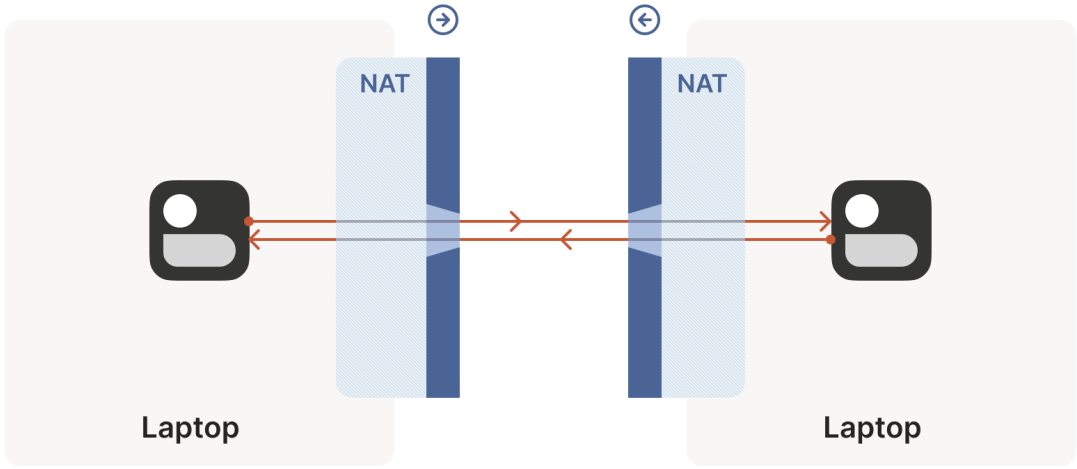
Traversal Using Relays around NAT

- Когда прямое соединение невозможно, все данные проксируются через сервер-ретранслятор.
- Relay – самый надежный, но и самый дорогой (латентность, нагрузка сервера)
- Примеры использования: WebRTC использует ICE: сначала пробует STUN/hole punching, потом переключается на TURN при необходимости

Продвинутые методы

- Birthday Paradox – статистический поиск совпадений портов
- Многопутевое пробирование портов
- Протоколы проброса портов (UPnP/NAT-PMP/PCP)
- Hairpinning при общих NAT
- Использование IPv6/NAT64
- ICE – активное пробование всех вариантов
- Relay-fallback – надёжная связь

Спасибо за внимание!



«Sharing is good, and with digital technology, sharing is easy[8].»

— Richard Matthew Stallman

[8] Especially if you have a good understanding of NATs...