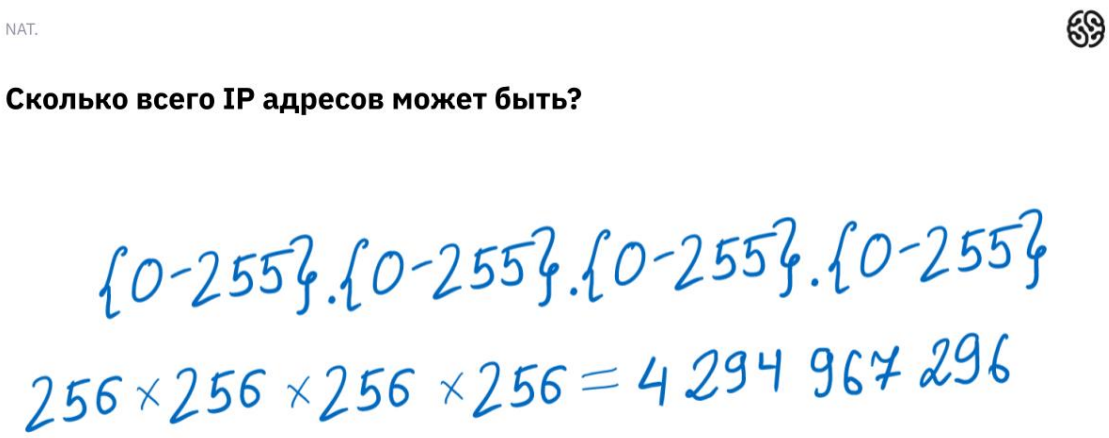
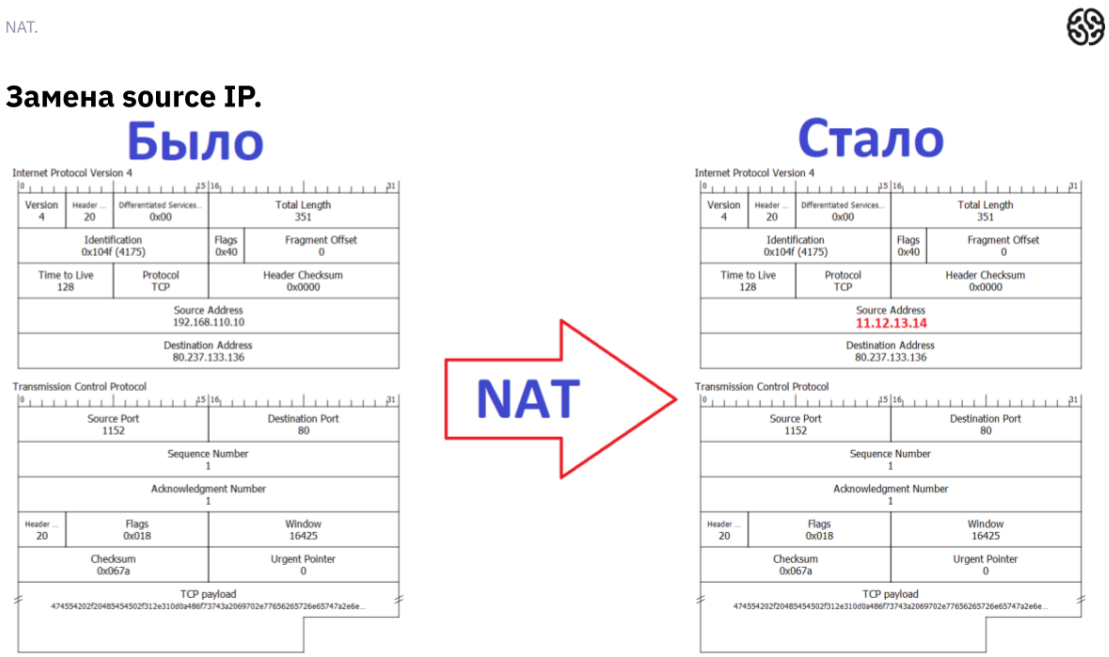
Лекция 5.

Уважаемые студенты, всем доброго времени суток! Я рад вас приветствовать на уже пятом уроке по компьютерным сетям. К этому уроку мы уже поняли что такое сети в целом, поняли как работает динамическая маршрутизация, поняли, что разделяя бродкаст-домены, мы можем строить большие сети, а также нам стало ясно, как доставляется траффик от приложения к приложению. Казалось бы, что нам известны все правила работы компьютерных сетей, но есть один нюанс.



Помните, когда мы изучали IP, я говорил, что каждый IP адрес состоит из 32 бит, разбитых по четыре октета, где в каждом октете 8 бит, то есть 256 значений, от 0 до 255? Из-за этого количество IP адресов ограничено числом 4 294 967 296. Это не очень много для современного мира, где почти у каждого человека на планете есть интернет. Но это было очень много в тот момент, когда Интернет только зарождался. И в этом плане протокол IP стал узким местом компьютерных сетей.

И сегодня мы будем говорить про технологию NAT, которая позволяет нам решать эту проблему, также познакомимся с понятиями публичных и приватных IP адресов и в конце начнем говорить про VPN.



Итак, на помощь к разрешению проблемы исчерпания IP адресов пришла технология NAT или Network Address Translation, с которой вам так или иначе придется столкнуться, а может кто-то уже и сталкивался. Суть технологии заключается в том, что она может менять IP адреса в заголовке пакета. Самый важный плюс этой технологии в том, что она позволяет экономить нам IP адреса. Именно благодаря этой технологии в мире интернетом может пользоваться больше чем 4 294 967 296 человек. Особенно учитывая, что интернет сегодня нужен не только компьютеру отдельно взятого человека, но также его смартфону, телевизору, умному чайнику, холодильнику, и так далее. Но просто замена IP с приватного на публичный не поможет нам в плане экономии IP адресов, нам надо пройти несколько шагов и усложнить эту простую замену на нашем слайде.

Private and Public IP.

Private IP address space	
From	**To**
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

Первое, что нам помогло избежать исчерпания IP адресов, это разделение их на два пула. Первый пул - это приватные IP адреса, еще их называют “серые” адреса. Это те адреса, которые используются у вас дома или у вас в офисе, или у вас в корпоративной сети. Эти IP адреса жёстко регламентированы, их диапазоны видны на слайде. Вы ими вправе пользоваться как вам угодно и вы можете распоряжаться ими как угодно, назначать на какие угодно хосты, но только в рамках своего LAN.

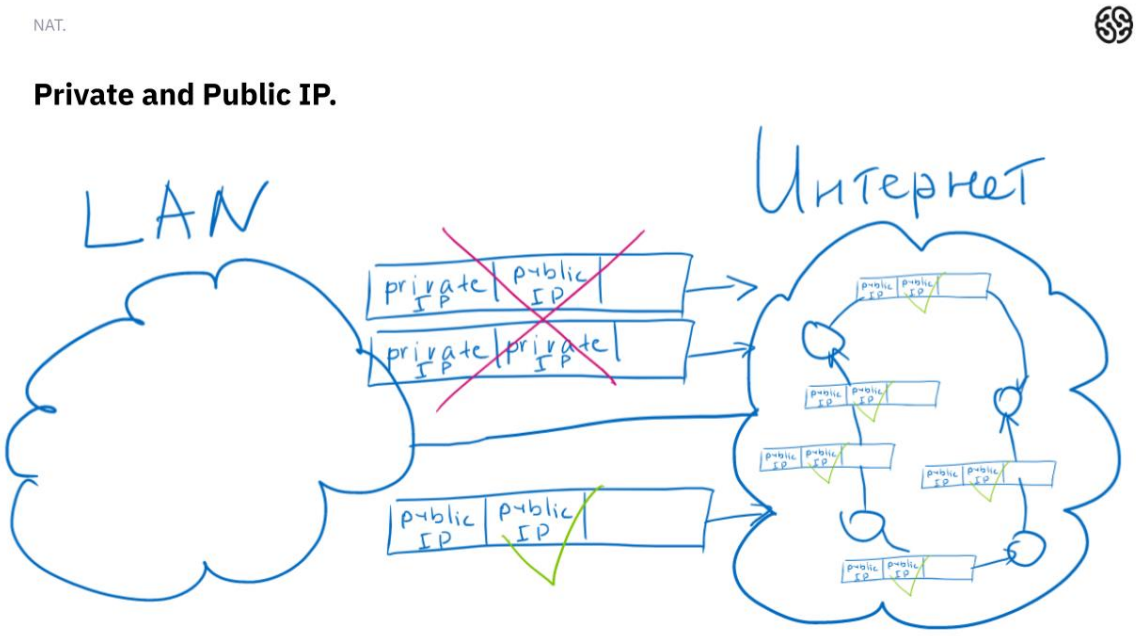
Второй пул - это пул так называемых публичных IP адресов или “белых” IP адресов. Это все IP адреса, кроме приватных и кроме специальных служебных адресов (помните, например IP 224.0.0.5 для OSPF? Вот это служебный IP).

Публичный IP может быть выдан для какой-либо организации, и этот публичный IP маршрутизируется в интернете, т.е. на всех роутерах в Интернете, у всех провайдеров, должен быть маршрут в сеть этого IP адреса. За распределение адресов между организациями отвечает специальная организация - IANA (Internet Assigned Numbers Authority). Она делегирует своим “дочерним” организациям -RIR (Regional Internet Registry) на континентах блоки адресов с маской /8.

В Европейском регионе RIR, который занимается выдачей IP адресов - это RIPE NCC (Réseaux IP Européens Network Coordination Centre).
В латиноамериканском регионе это LACNIC (Latin America and Caribbean Network Information Centre).
В африканском регионе AFRINIC (African Network Information Centre).
Ну и так далее.

Далее уже эти региональные представительства раздают блоки IP адресов с маской /24 и меньшей между юридическими организациями: крупными датацентрами, большими IT-компаниями, университетами, провайдерами, в том числе и вашему провайдеру. Благодаря наличию такой организации решается проблема, чтобы у двух организаций не было два одинаковых IP адреса во всём мире.

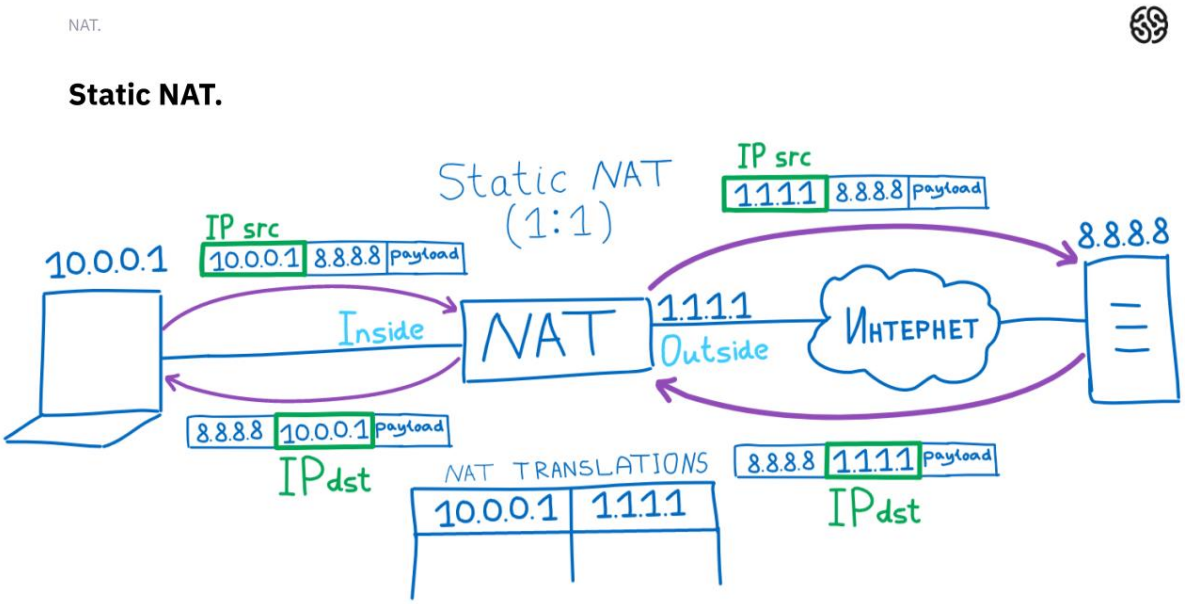
Так вот RIR’ы раздают как раз таки публичные IP адреса и трафик в интернете ходит только между публичными IP адресами.



Получается у нас как бы два мира айпи адресов. И два вида трафика: это ваш внутренний корпоративный трафик или внутренний домашний трафик, и трафик интернета.

В интернете по стандарту между провайдерами не должно ходить пакетов в которых Source IP или Destination IP будет с приватным IP адресом. Такой пакет по хорошему должен фильтроваться провайдером.

Поэтому в первую очередь нам нужно устройство которое транслирует приватные IP адреса в белые при прохождении трафика из LAN'а в интернет. И наоборот - транслировать белые IP адреса в ваши серые при прохождении трафика из интернета в LAN.



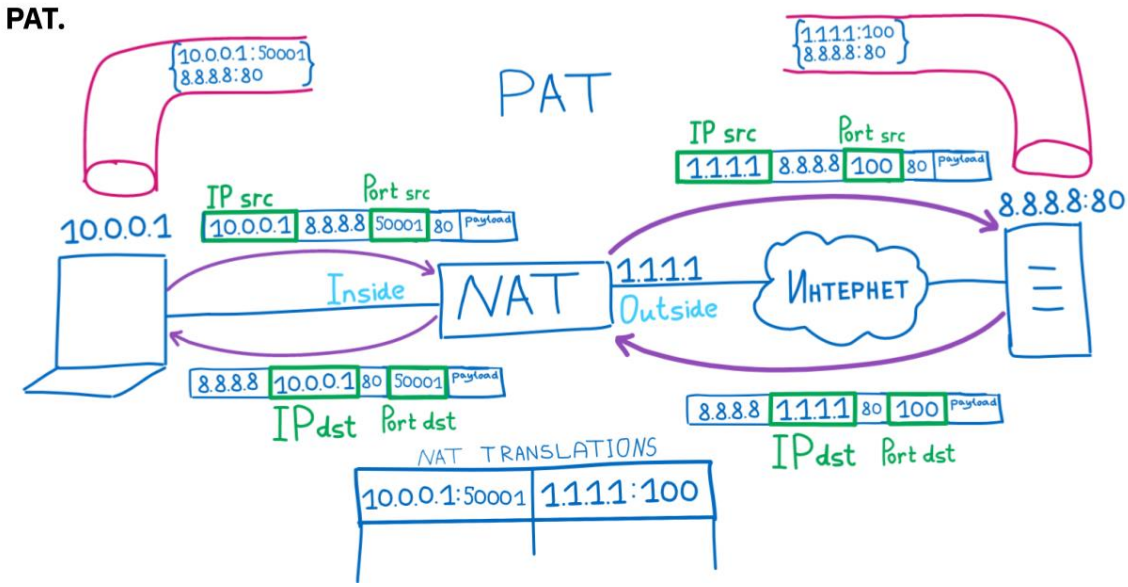
Сама трансляция называется NAT - Network Address Translation. Обычно она выполняется на роутерах, но может и выполняться и на отдельном устройстве или файерволе.

Представим что у нас есть один хост с приватным IP и ему надо в интернет. Давайте посмотрим как работает NAT в этом случае при трансляции 1 приватного IP адреса в один публичный IP адрес. На схеме у нас есть компьютер с IP 10.0.0.1, посередине NAT устройство, которое выполняет трансляцию и справа в интернете где-то есть веб-сервер с адресом 8.8.8.8. Рассмотрим по порядку, что происходит.

1. Итак, первое что делает компьютер - он создаёт пакет в сторону веб-сервера, ну например это будет syn-packet для создания tcp-сессии или просто udp пакет, неважно.

2. В устройстве NAT есть специальная таблица NAT трансляций. В данном случае, когда мы меняем 1 IP на 1 IP, в ней должна быть статическая (то есть внесенная руками админа) запись, что мы меняем 10.0.0.1 на 1.1.1.1. И когда на NAT пакет приходит от устройства на интерфейс с приватной стороны, NAT меняет Source IP адрес на адрес из таблицы, согласно записи. В нашем случае меняется IP 10.0.0.1 на 1.1.1.1. И такой пакет отправляется в интернет, в интернете пакет уже летит с Source IP адресом 1.1.1.1 на Destination IP адрес 8.8.8.8, то есть с публичного IP на публичный IP адрес.
3. Когда такой пакет долетает до нашего веб-сервера, он отвечает пакетом, в котором у нас уже Source IP адрес 8.8.8.8, а Destination IP адрес 1.1.1.1.
4. Когда ответный пакет доходит до нашего HAT устройства, оно меняет уже Destination(!) IP адрес, и вместо 1.1.1.1 подставляет 10.0.0.1. Опять таки согласно таблицы NAT.
5. Ответный пакет спокойно проходит через нашу LAN сеть до компьютера.

Таким образом происходит общение между компьютером в какой-то приватной сети и веб-сервером где-то в интернете. Важно понять, что компьютер даже не подозревает о том, что его пакеты где-то там NAT’ятся. Веб-сервер, получая пакеты, также не подозревает о каких-либо приватных IP адресах компьютера (ну, если вышестоящее приложение само не передает приватный IP адрес). NAT устройство таким образом “обманывает” всех, подменяя IP адреса. При этом NAT меняет Source IP при движении трафика с так называемого Inside интерфейса на Outside интерфейс. И наоборот, NAT меняет Destination IP адрес при движении трафика с Outside интерфейса на Inside.
Схема довольно прозрачная и понятная. Как видите ничего сложного здесь нет. Но мы можем заметить, что в таком случае у нас нет экономии публичных IP адресов. Если бы у нас добавился второй компьютер в нашей сети, то для того чтобы веб-сервер не перепутал трафик от одного компьютера до другого компьютера, нам необходим был бы второй публичный IP адрес, например 1.1.1.2. И вторая статическая запись в NAT таблице, согласно которой компьютер NAT’ился бы в этот IP - 1.1.1.2.



Давайте немного усложним нашу схему NAT трансляций с использованием заголовка L4. Такая схема называется Port Address Translation или PAT (встречаются также названия Overload NAT и маскарадинг). Это вид NAT, который как раз позволяет экономить айпи адреса.

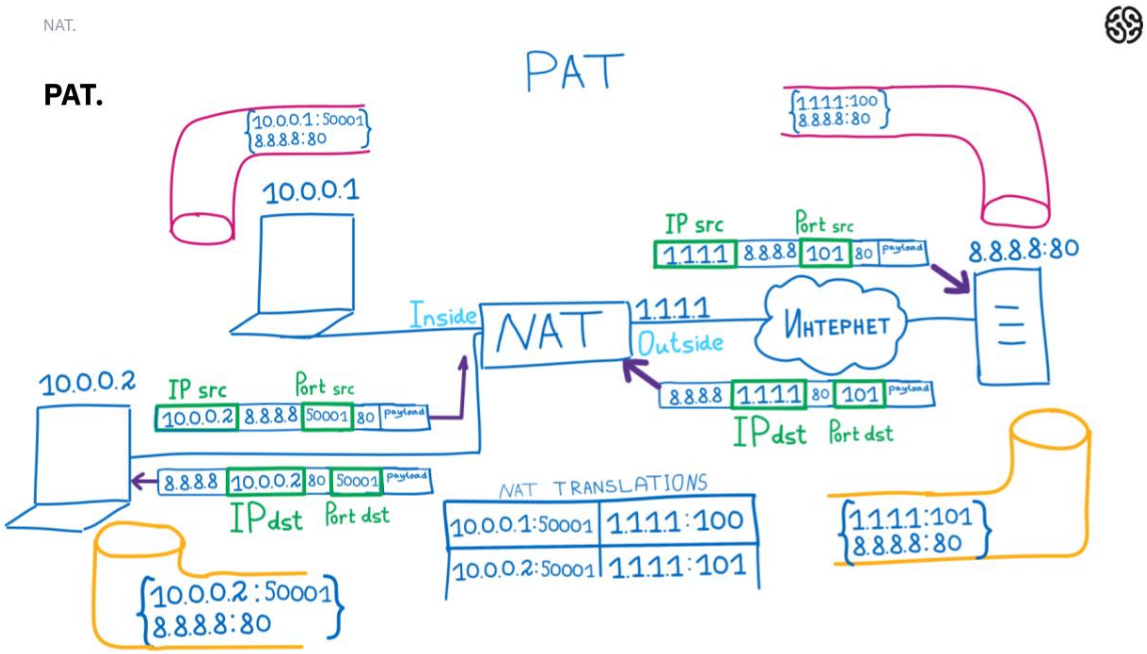
Предположим, что теперь у нас компьютер отправляет пакет на веб-сервер и в пакете следующая информация:

- Source IP адрес 10.0.0.1,
- Source port 50001,
- Destination IP адрес 8.8.8.8,
- Destination port 80.

1. Когда такой пакет проходит через PAT, в нем меняется Source IP адрес на 1.1.1.1, а также в нем меняется Source Port адрес, вместо 50001 например на 100.
2. Также в таблице NAT трансляций при этом появляется динамическая (то есть внесенная устройством автоматически) запись, что Source IP с адресом 10.0.0.1 и портом 50001, заменяется на 1.1.1.1 с портом 100.
3. Пакет дойдет до нашего веб-сервера где-то в интернете и на нем откроется socket с адресами 1.1.1.1:100 и 8.8.8.8:80.
4. Веб-сервер создаст ответный пакет, в котором уже Destination Address будет 1.1.1.1 и Destination Port 100. Когда пакет дойдет до нашего PAT, ему надо будет уже заменить Destination Address и Destination Port, на тот

который он найдет в своей таблице, в которой он заранее занёс запись. То есть, найдя запись 1.1.1.1:100 с 10.0.0.1:50001, мы меняем Destination Address на 10.0.0.1, а Destination Port на 50001.
5. Этот пакет уже дойдёт до нашего компьютера и спокойно обработается им в socket’е с адресами 10.0.0.1:50001 и 8.8.8.8:80.

Заметьте, здесь, как и в прошлый раз, компьютер ничего не знает о том что его пакеты где-то там проходят через NAT и в них меняется адресация. Веб-сервер также об этом ничего не знает.



6. Теперь давайте на нашей схеме добавим второй компьютер с адресом 10.0.0.2 и предположим, что он тоже стучится на веб-сервер на порт 80 со своего порта 50001, т.е. с такого же номера порта, с которого стучался компьютер №1. Ведь операционные системы этих компов никак не согласуют какие-то используемые порты между собой. Отправляя пакет на веб-сервер, у компа 2 также откроется socket, но с адресами 10.0.0.2:50001 и 8.8.8.8:80.

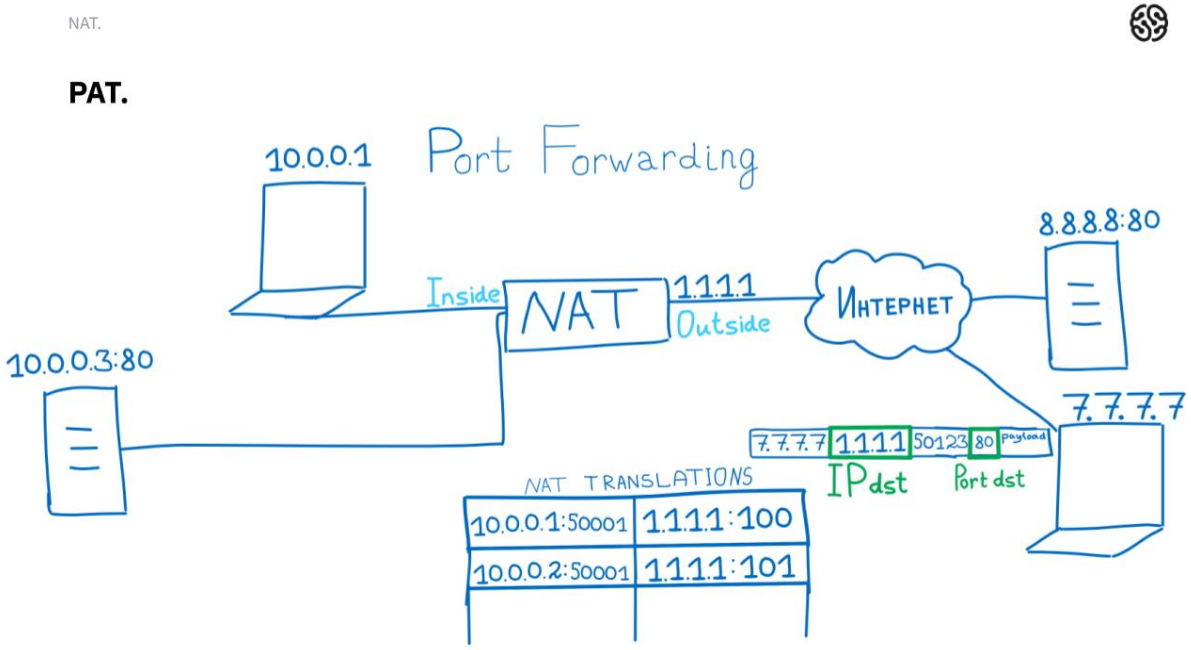
7. Дойдя до нашего устройства с NAT, оно поймет, что записи с этим компом и этим портом еще нет, и создаст такую запись, в которой IP:PORT 10.0.0.2:50001 заменяется на 1.1.1.1 со следующим номером порта 101.

8. До веб-сервера дойдет этот пакет и веб-сервер создаст новый socket со следующими параметрами 1.1.1.1:101 и 8.8.8.8:80. И отправит ответный пакет уже на Destination IP 1.1.1.1 и на Destination Port 80.

9. дойдя до нашего NAT, он произведет поиск в своей таблице трансляций и заменит соответствующей IP:PORT 1.1.1.1:101 на адрес 10.0.0.2 и порт 50001. Такой пакет дойдет до своего компьютера, который его отправил, и он обработает его в рамках своего сокета.

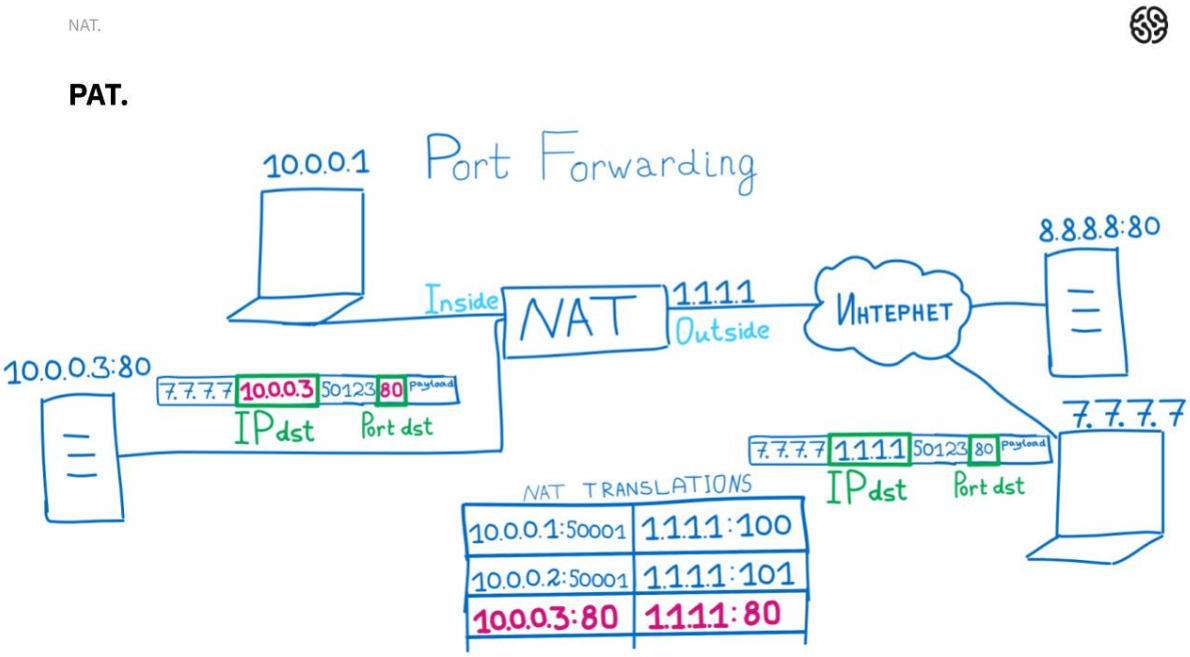
Таким образом мы за одним IP адресом можем скрыть кучу других IP адресов, заменяя порты в заголовках L4. С точки зрения сервера у нас просто много сессий открывается с одного IP 1.1.1.1, и не важно что есть устройство, которое скрывает за этим IP много других хостов.

Отсюда вытекает ограничение, что максимальное количество сессий может быть 65535, на один IP адрес. У нас может быть диапазон IP адресов, тогда и количество сессий может быть больше. NAT стремиться экономить сессии, поэтому если сессия закрывается корректно, то NAT сможет заново использовать освободившийся порт.



Теперь у нас всё будет работать хорошо до тех пор пока мы инициируем сессии с Inside стороны. Как только мы захотим разместить там, например, веб-сервер, то если кто-то захочет обратиться к нему из интернета, этот

кто-то будет слать к нам пакеты, в которых будет Destination IP 1.1.1.1 и Destination Port 80. Наш NAT к сожалению не найдёт записи с таким адресом и с таким портом. Поэтому для серверов мы должны делать отдельные статические записи, учитывая номер порта.



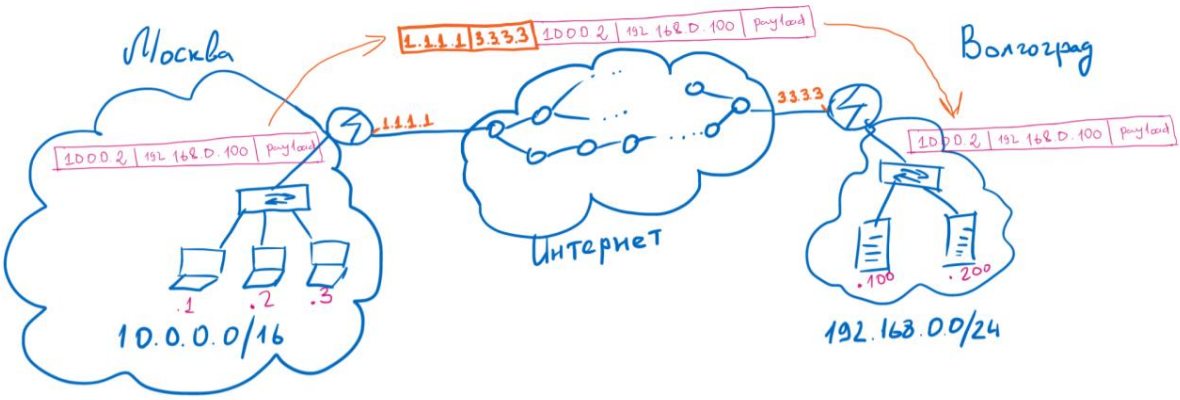
Например, для веб сервера с адресом 10.0.0.3 и портом 80, мы сделаем запись 1.1.1.1:80. Если помимо этого у нас будет FTP сервер с адресом 10.0.0.4 и портом 20, то и для него надо сделать запись с адресом 1.1.1.1:20. Таким образом мы можем размещать много серверов за одним публичным адресом, это так называемый Port forwarding или “проброс портов”.

Важно понимать что технология NAT ничего не знает о договорённости людей про публичные и приватные IP адреса. NAT трансляция может быть между любыми адресами, какими вам необходимо для решения своих задач. Все зависит только от того какие адреса вы настроите в правилах.

Про то, как настраивать NAT на роутера Cisco, мы более обстоятельно поговорим на семинаре. А сейчас давайте разберем еще один интересный вопрос - VPN!

VPN.

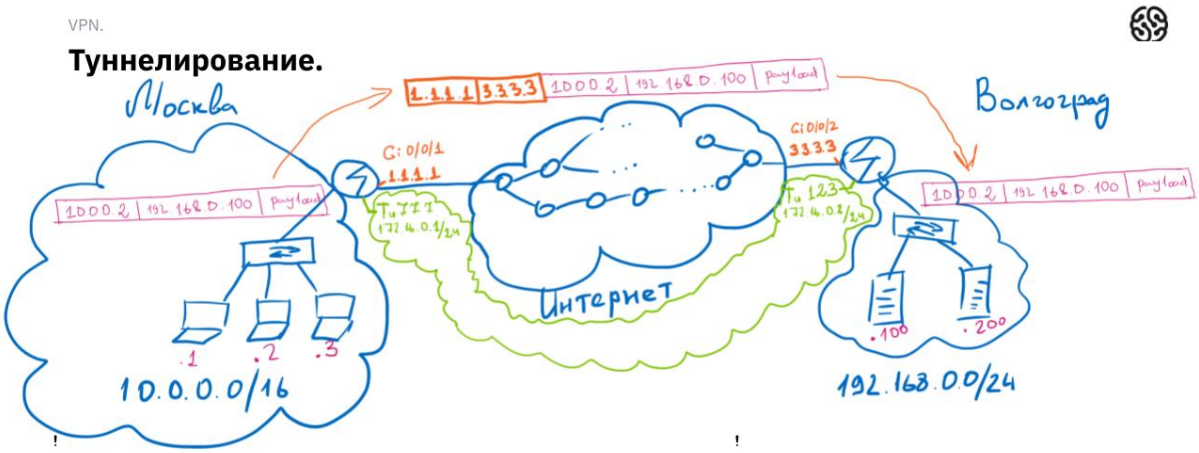
Туннелирование.



Итак, мы уже поняли, что наша локальная сеть может быть с приватными IP адресами, а в Интернет мы выходим с публичными. Но допустим, что у нас есть два офиса в разных городах и нам бы хотелось, чтобы хосты с одного офиса могли обращаться к серверам другого офиса, не выставляя сервера в интернет, т.е. не пробрасывая для них порты и не выделяя публичных адресов. Для решения такой задачи и существует технология VPN или Virtual Private Network.

Самих решений VPN очень много, но объединяет их всего лишь один простой принцип - туннелирование, которое мы рассмотрим на примере самого простого VPN - GRE (Generic Routing Encapsulation - общая инкапсуляция маршрутизации).

Итак, все на самом деле просто. Для того, чтобы наши пакеты между приватными сетями путешествовали по интернету, надо всего лишь… снабдить пакет еще одним заголовком IP, в котором будут публичные адреса наших филиалов. Законом не запрещено. Ведь роутеры не смотрят весь пакет, они смотрят только L2 заголовок и следующий за ним L3 заголовок, на основе которого и пересылают пакет дальше. Таким образом мы как бы обманываем все роутеры в интернете, это и называется туннелированием. Получается как бы что наш пакет с приватными заголовками въезжает в туннель на роутере в первом филиале, и выезжает из роутера на втором филиале. А интернет - гора.




```
!
interface Tunnel777
 ip address 172.16.0.1 255.255.255.0
 mtu 1476
 tunnel source GigabitEthernet0/0/1
 tunnel destination 3.3.3.3
!
ip route 192.168.0.0 255.255.255.0 172.16.0.2
!
```

```
!
interface Tunnel123
 ip address 172.16.0.2 255.255.255.0
 mtu 1476
 tunnel source GigabitEthernet0/0/2
 tunnel destination 1.1.1.1
!
ip route 10.0.0.0 255.255.0.0 172.16.0.1
!
```

Рассмотрим пример более подробно. Есть у нас офис в Москве и филиал в Волгограде. Допустим приватная сеть в Москве - 10.0.0.0/16. А приватная сеть в Волгограде - 192.168.0.0/24. Для того, чтобы настроить туннель и соединить эти сети через VPN, на роутерах необходимо настроить туннельный интерфейс. Это почти как обычный физический L3 интерфейс, только не привязанный ни к какому физическому порту. На нем настраивается IP адрес из вспомогательной сети. Например 172.16.0.1/24 на роутере в Москве и 172.16.0.2/24 на роутере в Волгограде. Теперь, прописав в Москве статический маршрут что сеть 192.168.0.0/24 за 172.16.0.2/24 , а в Волгограде прописав 10.0.0.0/16 за 172.16.0.1/24. Мы свяжем эти сети. Представим, что у нас идёт пакет из Москвы с Source IP например 10.0.0.2 на какой-нибудь сервер в Волгограде, например на Destination IP 192.168.0.100. Как только такой пакет дойдет до роутера, первым делом он по таблице маршрутизации увидит, что IP 192.168.0.2 надо отправить на 172.16.0.2/24, которая у нас “Connected” к интерфейсу Tunnel777. Т.к. это интерфейс типа Tunnel. Cisco навесит заголовок GRE плюс еще один заголовок IP, в котором IP Source и IP Destination будут те, которые указаны в настройках нашего интерфейса Tunnel777. Т.е. там будут публичные адреса, которые без проблем смаршрутизируются в Интернете. Такой пакет, когда придёт на роутер в Волгограде, будет обработан самим роутером. Он увидит, что Destination IP там он сам, и что там GRE, далее он отрежет лишний L3 заголовок. Затем посмотрит на Destination IP адрес в оставшемся IP заголовке. Увидит там 192.168.0.100 и

по таблице роутинга поймет что это сеть Connected и отправит дальше уже непосредственно на сервер. Ни хост в Москве, ни сервер в Волгограде не знают ничего о том, что их пакеты где-то туннелируются, этим занимаются роутеры. С точки зрения друг друга, они находятся в одной корпоративной среде. В этом и суть любого VPN соединения.

Стоить отметить, что GRE увеличивает размер пакета. И если у нас будет снабжен дополнительным заголовком пакет, в котором payload 1500 байт. В интернет он вылетит уже с payload равным 1524 байта, где +20 байт заголовок IP и + 4 байта заголовок GRE. Из второй лекции мы должны помнить, что стандарт де факто в интернете для MTU (Maximum Transmit Unit - т.е. максимальный размер payload'а для L2) равен 1500 байтам. И чтобы пакет не стал размером больше 1500 байт, и не был отброшен провайдерами в Интернете, на туннельном интерфейсе специально автоматически прописывается MTU 1476 байт. Т.е. из 1500 мы вычитаем 20 байт на заголовок IP и еще 4 байта занимает дополнительный заголовок GRE. Но и на своих клиентах рекомендуется уменьшить MTU до 1476 байтов.

У такого соединения через GRE, есть один существенный недостаток - если кто-то в интернете перехватит ваш трафик, он без проблем поймет с какими адресами и с каким payload'ом у вас общаются хосты между собой в вашей сети. Поэтому такие пакеты желательно шифровать и защищать. Но о шифровании и безопасности мы поговорим на следующем уроке.

А сегодня мы изучили важную тему трансляций сетевых адресов - NAT, поговорили о самых используемых его видах, а также поняли основной принцип по которому работает VPN - туннелирование. На семинаре мы как раз попрактикуемся с настройкой NAT и GRE. Приходите, будет интересно. Ну а на сегодня все, до следующего занятия!