19. Динамические системы именования. Принципы организации DNS. Рекурсивные и итеративные запросы. Выполнение DNS-запросов через интерфейс прикладного программирования Sockets.

Динамические системы именования

DHCP(DynamicHostControlProtocol) – специальный протокол.

Компьютер, входя в сеть, посылает широковещательный пакет, что бы узнать, какие есть в сети DHCP-сервера. Один из них откликается, отсылая IP, который должен принять компьютер. На DHCP-сервере хранится список занятых IP, и зачастую таблица соответствия МАС-IP, что бы при переконфигурировании выдавать узлу тот жеIP. Так же узлу задается и маска и основной шлюз (сервер можно настроить так, что бы определенным МАС-ам выдавались определенные IP). То есть становится возможным автоматическое задание настроек сети.

Принципы организации dns. Рекурсивные и итеративные запросы.

Домены первого уровня администрируются единой организацией, находящейся в Америке.

При задании адреса перечисляется вся иерархическая цепочка от вершины к корню, разделяя доменные имена точкой.

DNS не чувствителен к регистру.

Владелец домена управляет всеми префиксами.

Система префиксная в силу того, что человек лучше запоминает префикс.

DNS фактически является распределенной базой данных.

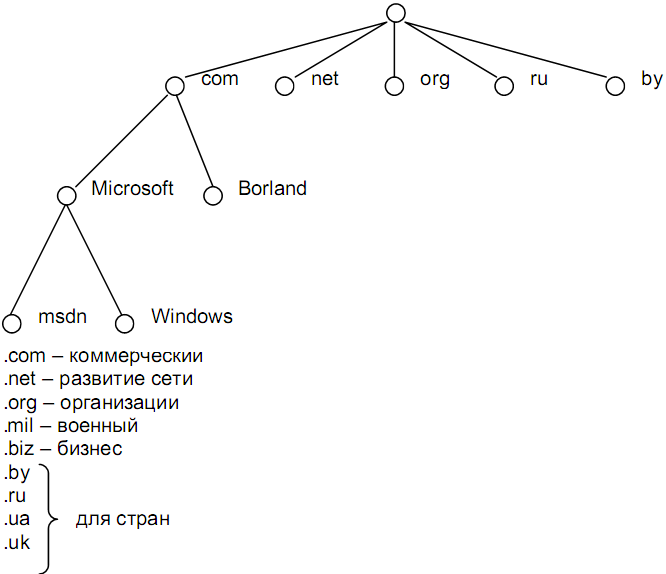
DNS используется протоколами, работающими над ТСР.

Система DNS стандартизирует форматы имен и запросы на получение каких-то данных, ассоциирующихся с этими именами. Все имена объединяются в иерархию, представляющую собой дерево.

Имена обладают структурой, они составные.

Корнем дерева имен является пустое имя, имя без единого символа.

Добавляя к нему через точку префиксы, мы формируем уточненное имя, соответствующее некоторому узлу в глобальном дереве имен.



Каждый узел называют доменом.

Корень – нулевой уровень.

Имена, или домены первого уровня стандартизированы. Добавлением имен первого уровня занимается одна глобальная организация. На настоящее время они уже не добавляются.

Дерево можно нарезать на зоны. Зона отдается организации, которая занимается ее расширением. Делается это для того, чтобы куски дерева могли находиться на разных компьютерах.

msdn.microsoft.com

К этому имени могут быть подключены записи различного типа (IP, email).

- A (address) – записи об IP-адресах

DNS используется для хранения IP-адресов.

Рассмотрим, как выполняются DNS-запросы.

Они бывают двух видов:

- итеративные

- рекурсивные

**Итеративный запрос.**

На каждом компьютере, который подключен к Internet и DNS, имеется DNS-агент. Он ищет в своей локальной таблице соответствий искомое имя. Если нет, то он обращается с запросом к своему DNS-серверу верхнего уровня. DNS-сервер ищет у себя и возвращает либо результат, либо признак «не найден» + адрес другого DNS-сервера, который рекомендуется для продолжения DNS-запроса. Таким образом, в этом режиме DNS-агент выполняет перебор DNS-серверов сам, пока не будет найден ответ.

Рассмотрим пример:

DNS-агент попытается обратиться к серверу .com, передав ему имя msdn.microsoft.com. Если в домене верхнего уровня этот адрес явно не прописан (а он не прописан), сервер возвратит «не знаю» + IP-адрес сервера microsoft.com. DNS-агент обратится к microsoft.com, опросит его. На этом сервере хранится IP-адрес того DNS-сервера, на котором было зарегистрировано это имя.

Возвратит «не найдено» + IP msdn.microsoft.com.

Перебор по циклу выполняет DNS-агент.

Адрес DNS-сервера задается в свойствах протокола (ТСР) или может устанавливаться автоматически (поDHCP).

**Рекурсивный запрос.**

Заключается в том, что DNS-агент спрашивает DNS-сервер: «Узнай мне IP такого имени».

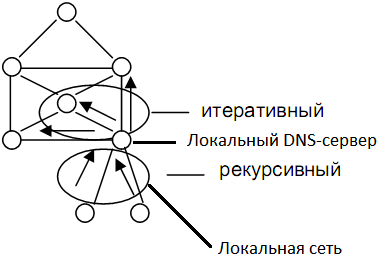
DNS-сервер сам занимается поиском и возвращает конечный результат. То есть делегируются обязанности.

Рекурсивный запрос лучше тем, что возникает возможность кэширования адресов (ведь поиск все время занимает сервер, а не клиентские машины и клиентам ищущим один и тот же адрес, можно сразу дать ответ).Если бы каждый клиент сам искал адрес итеративно то это сильно бы загружало сеть этими поисковыми запросами.

Однако, если бы все испоьзовали рекурсивный поиск, то все DNS-сервера пытались бы спихнуть на более выше стоящиеDNS-сервера и на центральныеDNS-сервера приходилось бы просто гигантская нагрузка.

Поэтому, на практике делается так:

+Терминальные узлы в локальной сети используют рекурсивные запросы, а уже DNS-сервера локальных сетей используют итеративные запросы (то есть сами ищут адреса никому не перепоручая это задачу)



Таким образом, количество серверов, занимающихся поиском IP адресов, достаточно невелико, что бы не перегружать сеть, и, в то же время, их достаточно много, что бы справляться с нагрузкой (не идет перегрузки на центральные сервера).

Очень опасны атаки на DNS-сервера, так как просто переписав соответствие между символьным именем и IP-адресом, мы переправим все запросы этого ресурса на другой ресурс. Если же DNS-сервера лягут, то станет невозможным использование каких-либо интернет услуг.

20. Понятие сетевого экрана. Основные принципы его функционирования.

Сетевой экран – это специализированное ПО, призванное обеспечить защиту находящихся за ним узлов от атак извне.

Существуют сетевые экраны, защищающие локальную сеть от угроз из внешней сети (интернет). Это обычно выделенный сервер.

Локальные сетевые экраны ставятся на конкретный узел и защищают узел от угроз, как из интернета, так и из локальной сети.

Существуют экраны работающие на сетевом, на транспортном, на канальном и на прикладном уровне. Иногда работают на нескольких уровнях сразу.

Функции:

1. Разрешение и запрет трафика – это основная детерминированная задача.
2. Анализ трафика

Трафик делится на исходящий и входящий.

В протоколе ТСР для передачи данных надо создать соединение. Кто-то его устанавливает (инициирует), а кто-то подтверждает. Если узел создает соединение, то экран обычно разрешает исходящий трафик и запрещает входящий, т.е. не дает кому-то извне инициализировать соединение с компом.

Для протокола UDP установлено такое соглашение:

Первый по времени пакет от А с порта ПА на узел Б на порт ПБ считается запросом на установление соединения. Далее все пакеты с ПА на ПБ и с ПБ на ПА разрешаются.

Возникает вопрос, что если соединение логически закончилось и начинается новое с ПБ на ПА, то это уже нелегальный пакет. На такой случай в экране предусмотрен таймаут, по прохождении которого соединение считается оборванным.

Так работает экран на транспортном уровне.

+На канальном уровне можно запретить передачу пакетов от узлов с заданными МАС-адресами.

21. Технология NAT. Принципы организации и функционирования. Статическая и динамическая трансляция адресов и портов (PAT).

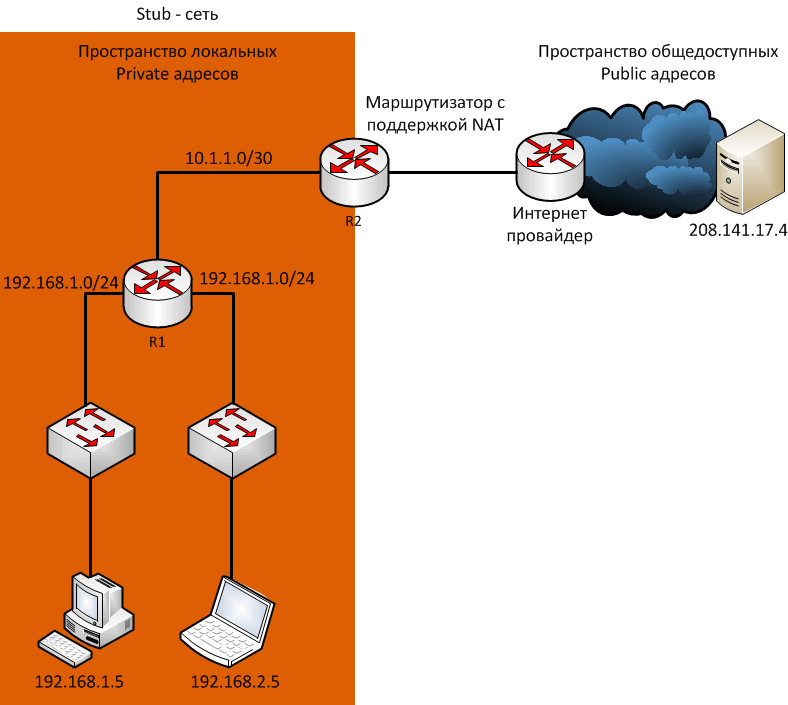
232 или 4 294 967 296 **IPv4** адресов это много? Кажется, что да. Однако с распространением персональных вычислений, мобильных устройств и быстрым ростом интернета вскоре стало очевидно, что 4,3 миллиарда адресов IPv4 будет недостаточно. Долгосрочным решением было **IPv6**, но требовались более быстрое решение для устранения нехватки адресов. И этим решением стал **NAT (Network Address Translation)**.

Сети обычно проектируются с использованием частных IP адресов. Это адреса **10.0.0.0/8, 172.16.0.0/12** и **192.168.0.0/16**. Эти частные адреса используются внутри организации или площадки, чтобы позволить устройствам общаться локально, и они не маршрутизируются в интернете. Чтобы позволить устройству с приватным IPv4-адресом обращаться к устройствам и ресурсам за пределами локальной сети, приватный адрес сначала должен быть переведен на общедоступный публичный адрес.

И вот как раз NAT переводит приватные адреса, в общедоступные. Это позволяет устройству с частным адресом IPv4 обращаться к ресурсам за пределами его частной сети. NAT в сочетании с частными адресами IPv4 оказался полезным методом сохранения общедоступных IPv4-адресов. Один общедоступный IPv4-адрес может быть использован сотнями, даже тысячами устройств, каждый из которых имеет частный IPv4-адрес. NAT имеет дополнительное преимущество, заключающееся в добавлении степени конфиденциальности и безопасности в сеть, поскольку он скрывает внутренние IPv4-адреса из внешних сетей.

Маршрутизаторы с поддержкой NAT могут быть настроены с одним или несколькими действительными общедоступными IPv4-адресами. Эти общедоступные адреса называются пулом NAT. Когда устройство из внутренней сети отправляет трафик из сети наружу, то маршрутизатор с поддержкой NAT переводит внутренний IPv4-адрес устройства на общедоступный адрес из пула NAT. Для внешних устройств весь трафик, входящий и выходящий из сети, выглядит имеющим общедоступный IPv4 адрес.

Маршрутизатор NAT обычно работает на границе **Stub**-сети. Stub-сеть – это тупиковая сеть, которая имеет одно соединение с соседней сетью, один вход и выход из сети.



Когда устройство внутри Stub-сети хочет связываться с устройством за пределами своей сети, пакет пересылается пограничному маршрутизатору, и он выполняет NAT-процесс, переводя внутренний частный адрес устройства на публичный, внешний, маршрутизируемый адрес.

**ТЕРМИНОЛОГИЯ NAT**

В терминологии NAT внутренняя сеть представляет собой набор сетей, подлежащих переводу. Внешняя сеть относится ко всем другим сетям.

При использовании NAT, адреса IPv4 имеют разные обозначения, основанные на том, находятся ли они в частной сети или в общедоступной сети (в интернете), и является ли трафик входящим или исходящим.

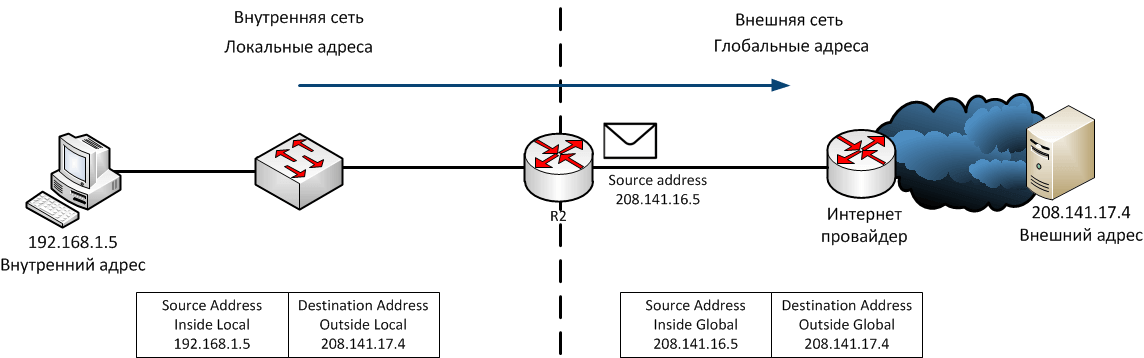
NAT включает в себя четыре типа адресов:

* **Внутренний локальный адрес (Inside local address)**;
* **Внутренний глобальный адрес (Inside global address)**;
* **Внешний местный адрес (Outside local address)**;
* **Внешний глобальный адрес (Outside global address)**;

При определении того, какой тип адреса используется, важно помнить, что терминология NAT всегда применяется с точки зрения устройства с транслированным адресом:

* **Внутренний адрес (Inside address)** - адрес устройства, которое транслируется NAT;
* **Внешний адрес (Outside address)** - адрес устройства назначения;
* **Локальный адрес (Local address)** - это любой адрес, который отображается во внутренней части сети;
* **Глобальный адрес (Global address)** - это любой адрес, который отображается во внешней части сети;

Рассмотрим это на примере схемы.

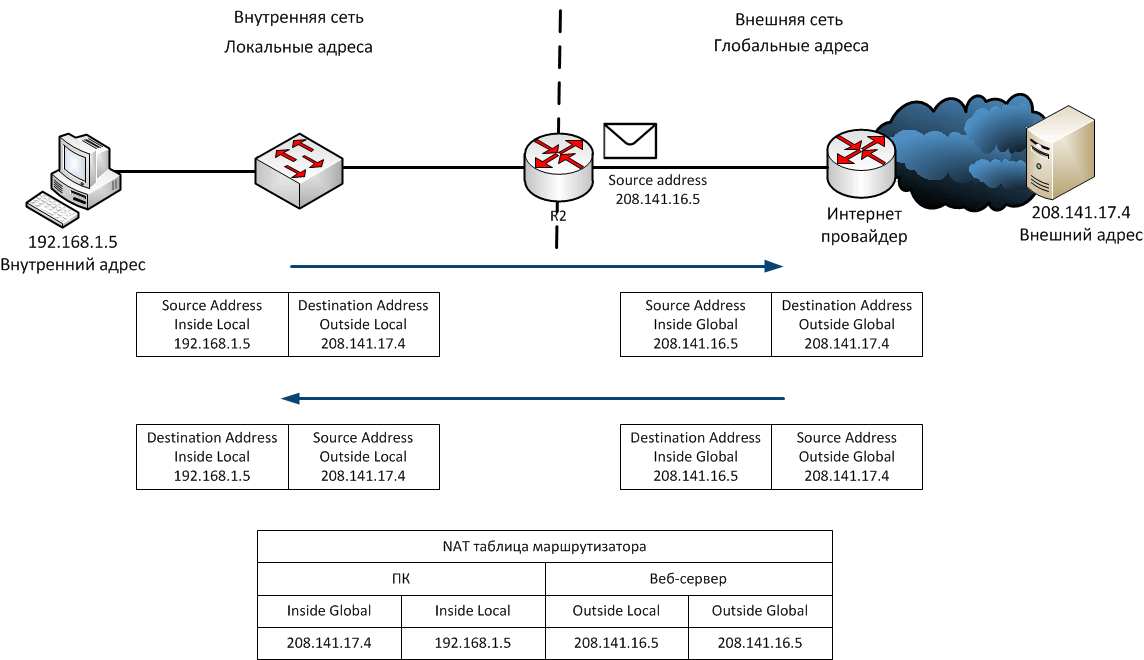


На рисунке ПК имеет внутренний локальный (**Inside local**) адрес 192.168.1.5 и с его точки зрения веб-сервер имеет внешний (**outside**) адрес 208.141.17.4. Когда с ПК отправляются пакеты на глобальный адрес веб-сервера, внутренний локальный (**Inside local**) адрес ПК транслируется в 208.141.16.5 (**inside global**). Адрес внешнего устройства обычно не переводится, поскольку он является общедоступным адресом IPv4.

Стоит заметить, что ПК имеет разные локальные и глобальные адреса, тогда как веб-сервер имеет одинаковый публичный IP адрес. С его точки зрения трафик, исходящий из ПК поступает с внутреннего глобального адреса 208.141.16.5. Маршрутизатор с NAT является точкой демаркации между внутренней и внешней сетями и между локальными и глобальными адресами.

Термины, **inside** и **outside**, объединены с терминами **local** и**global**, чтобы ссылаться на конкретные адреса. На рисунке маршрутизатор настроен на предоставление NAT и имеет пул общедоступных адресов для назначения внутренним хостам.

На рисунке показано как трафик отправляется с внутреннего ПК на внешний веб-сервер, через маршрутизатор с поддержкой NAT, и высылается и переводится в обратную сторону.



Внутренний локальный адрес (**Inside local address**) - адрес источника, видимый из внутренней сети. На рисунке адрес 192.168.1.5 присвоен ПК – это и есть его внутренний локальный адрес.

Внутренний глобальный адрес (**Inside global address**) - адрес источника, видимый из внешней сети. На рисунке, когда трафик с ПК отправляется на веб-сервер по адресу 208.141.17.4, маршрутизатор переводит внутренний локальный адрес (**Inside local address**) на внутренний глобальный адрес (**Inside global address**). В этом случае роутер изменяет адрес источника IPv4 с 192.168.1.5 на 208.141.16.5.

Внешний глобальный адрес (**Outside global address**) - адрес адресата, видимый из внешней сети. Это глобально маршрутизируемый IPv4-адрес, назначенный хосту в Интернете. На схеме веб-сервер доступен по адресу 208.141.17.4. Чаще всего внешние локальные и внешние глобальные адреса одинаковы.

Внешний локальный адрес (**Outside local address**) - адрес получателя, видимый из внутренней сети. В этом примере ПК отправляет трафик на веб-сервер по адресу 208.141.17.4

Рассмотрим весь путь прохождения пакета. ПК с адресом 192.168.1.5 пытается установить связь с веб-сервером 208.141.17.4. Когда пакет прибывает в маршрутизатор с поддержкой NAT, он считывает IPv4 адрес назначения пакета, чтобы определить, соответствует ли пакет критериям, указанным для перевода. В этом пример исходный адрес соответствует критериям и переводится с 192.168.1.5 (**Inside local address**) на 208.141.16.5. (**Inside global address**). Роутер добавляет это сопоставление локального в глобальный адрес в таблицу NAT и отправляет пакет с переведенным адресом источника в пункт назначения. Веб-сервер отвечает пакетом, адресованным внутреннему глобальному адресу ПК (208.141.16.5). Роутер получает пакет с адресом назначения 208.141.16.5 и проверяет таблицу NAT, в которой находит запись для этого сопоставления. Он использует эту информацию и переводит обратно внутренний глобальный адрес (208.141.16.5) на внутренний локальный адрес (192.168.1.5), и пакет перенаправляется в сторону ПК.

**ТИПЫ NAT**

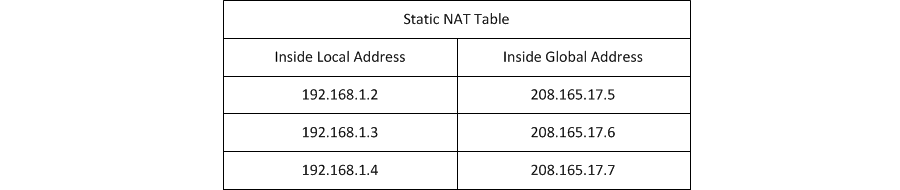
Существует три типа трансляции NAT:

* **Статическая адресная трансляция (Static NAT)** - сопоставление адресов один к одному между локальными и глобальными адресами;
* **Динамическая адресная трансляция (Dynamic NAT)** - сопоставление адресов “многие ко многим” между локальными и глобальными адресами;
* **Port Address Translation (NAT)** - многоадресное сопоставление адресов между локальными и глобальными адресами c использованием портов. Также этот метод известен как **NAT Overload**;

STATIC NAT

Статический NAT использует сопоставление локальных и глобальных адресов один к одному. Эти сопоставления настраиваются администратором сети и остаются постоянными. Когда устройства отправляют трафик в Интернет, их внутренние локальные адреса переводятся в настроенные внутренние глобальные адреса. Для внешних сетей эти устройства имеют общедоступные IPv4-адреса. Статический NAT особенно полезен для веб-серверов или устройств, которые должны иметь согласованный адрес, доступный из Интернета, как например веб-сервер компании. Статический NAT требует наличия достаточного количества общедоступных адресов для удовлетворения общего количества одновременных сеансов пользователя.

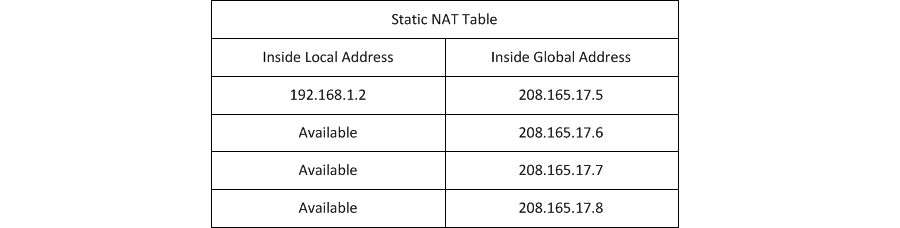
Статическая NAT таблица выглядит так:



**DYNAMIC NAT**

Динамический NAT использует пул публичных адресов и назначает их по принципу «первым пришел, первым обслужен». Когда внутреннее устройство запрашивает доступ к внешней сети, динамический NAT назначает доступный общедоступный IPv4-адрес из пула. Подобно статическому NAT, динамический NAT требует наличия достаточного количества общедоступных адресов для удовлетворения общего количества одновременных сеансов пользователя.

Динамическая NAT таблица выглядит так:

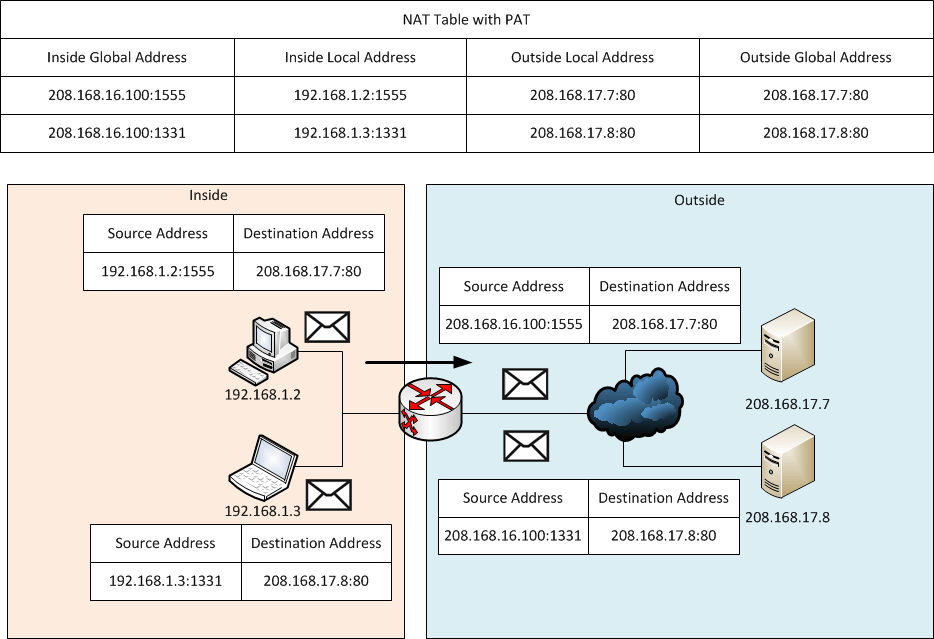


**PORT ADDRESS TRANSLATION (PAT)**

**PAT** транслирует несколько частных адресов на один или несколько общедоступных адресов. Это то, что делают большинство домашних маршрутизаторов. Интернет-провайдер назначает один адрес маршрутизатору, но несколько членов семьи могут одновременно получать доступ к Интернету. Это наиболее распространенная форма NAT.

С помощью PAT несколько адресов могут быть сопоставлены с одним или несколькими адресами, поскольку каждый частный адрес также отслеживается номером порта. Когда устройство инициирует сеанс **TCP/IP**, оно генерирует значение порта источника **TCP** или **UDP** для уникальной идентификации сеанса. Когда NAT-маршрутизатор получает пакет от клиента, он использует номер своего исходного порта, чтобы однозначно идентифицировать конкретный перевод NAT. PAT гарантирует, что устройства используют разный номер порта TCP для каждого сеанса. Когда ответ возвращается с сервера, номер порта источника, который становится номером порта назначения в обратном пути, определяет, какое устройство маршрутизатор перенаправляет пакеты.

Картинка иллюстрирует процесс PAT. PAT добавляет уникальные номера портов источника во внутренний глобальный адрес, чтобы различать переводы.



Поскольку маршрутизатор обрабатывает каждый пакет, он использует номер порта (1331 и 1555, в этом примере), чтобы идентифицировать устройство, с которого выслан пакет.

Адрес источника (**Source Address**) - это внутренний локальный адрес с добавленным номером порта, назначенным TCP/IP. Адрес назначения (**Destination Address**) - это внешний локальный адрес с добавленным номером служебного порта. В этом примере порт службы 80: HTTP.

Для исходного адреса маршрутизатор переводит внутренний локальный адрес во внутренний глобальный адрес с добавленным номером порта. Адрес назначения не изменяется, но теперь он называется внешним глобальным IP-адресом. Когда веб-сервер отвечает, путь обратный.

В этом примере номера портов клиента 1331 и 1555 не изменялись на маршрутизаторе с NAT. Это не очень вероятный сценарий, потому что есть хорошая вероятность того, что эти номера портов уже были прикреплены к другим активным сеансам. PAT пытается сохранить исходный порт источника. Однако, если исходный порт источника уже используется, PAT назначает первый доступный номер порта, начиная с начала соответствующей группы портов **0-511, 512-1023** или **1024-65535**. Когда портов больше нет, и в пуле адресов имеется более одного внешнего адреса, PAT переходит на следующий адрес, чтобы попытаться выделить исходный порт источника. Этот процесс продолжается до тех пор, пока не будет доступных портов или внешних IP-адресов.

То есть если другой хост может выбрать тот же номер порта 1444. Это приемлемо для внутреннего адреса, потому что хосты имеют уникальные частные IP-адреса. Однако на маршрутизаторе NAT номера портов должны быть изменены - в противном случае пакеты из двух разных хостов выйдут из него с тем же адресом источника. Поэтому PAT назначает следующий доступный порт (1445) на второй адрес хоста.

Подведем итоги в сравнении NAT и PAT. Как видно из таблиц, NAT переводит IPv4-адреса на основе 1:1 между частными адресами IPv4 и общедоступными IPv4-адресами. Однако PAT изменяет как сам адрес, так и номер порта. NAT перенаправляет входящие пакеты на их внутренний адрес, ориентируясь на входящий IP адрес источника, заданный хостом в общедоступной сети, а с PAT обычно имеется только один или очень мало публично открытых IPv4-адресов, и входящие пакеты перенаправляются, ориентируясь на NAT таблицу маршрутизатора.

А что относительно пакетов IPv4, содержащих данные, отличные от TCP или UDP? Эти пакеты не содержат номер порта уровня 4. PAT переводит наиболее распространенные протоколы, переносимые IPv4, которые не используют TCP или UDP в качестве протокола транспортного уровня. Наиболее распространенными из них являются ICMPv4. Каждый из этих типов протоколов по-разному обрабатывается PAT. Например, сообщения запроса ICMPv4, эхо-запросы и ответы включают идентификатор запроса **Query ID**. ICMPv4 использует Query ID. для идентификации эхо-запроса с соответствующим ответом. Идентификатор запроса увеличивается с каждым отправленным эхо-запросом. PAT использует идентификатор запроса вместо номера порта уровня 4.

**ПРЕИМУЩЕСТВА И НЕДОСТАТКИ NAT**

NAT предоставляет множество преимуществ, в том числе:

* NAT сохраняет зарегистрированную схему адресации, разрешая приватизацию интрасетей. При PAT внутренние хосты могут совместно использовать один общедоступный IPv4-адрес для всех внешних коммуникаций. В этом типе конфигурации требуется очень мало внешних адресов для поддержки многих внутренних хостов;
* NAT повышает гибкость соединений с общедоступной сетью. Многочисленные пулы, пулы резервного копирования и пулы балансировки нагрузки могут быть реализованы для обеспечения надежных общедоступных сетевых подключений;

NAT обеспечивает согласованность для внутренних схем адресации сети. В сети, не использующей частные IPv4-адреса и NAT, изменение общей схемы адресов IPv4 требует переадресации всех хостов в существующей сети. Стоимость переадресации хостов может быть значительной. NAT позволяет существующей частной адресной схеме IPv4 оставаться, позволяя легко изменять новую схему общедоступной адресации. Это означает, что организация может менять провайдеров и не нужно менять ни одного из своих внутренних клиентов;

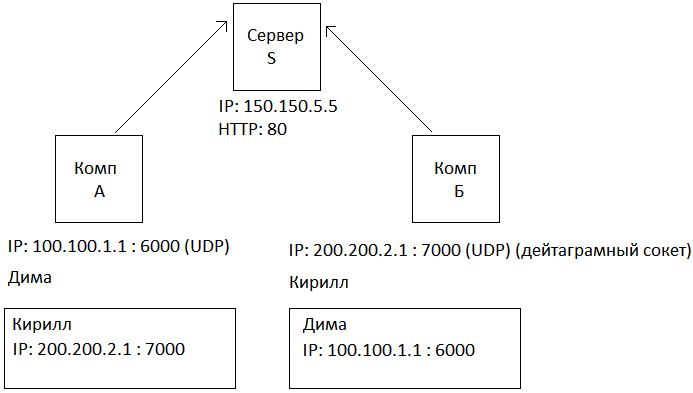
* NAT обеспечивает сетевую безопасность. Поскольку частные сети не рекламируют свои адреса или внутреннюю топологию, они остаются достаточно надежными при использовании в сочетании с NAT для получения контролируемого внешнего доступа. Однако нужно понимать, что NAT не заменяет фаерволы;

Но у NAT есть некоторые недостатки. Тот факт, что хосты в Интернете, по-видимому, напрямую взаимодействуют с устройством с поддержкой NAT, а не с фактическим хостом внутри частной сети, создает ряд проблем:

* Один из недостатков использования NAT связан с производительностью сети, особенно для протоколов реального времени, таких как **VoIP**. NAT увеличивает задержки переключения, потому что перевод каждого адреса IPv4 в заголовках пакетов требует времени;
* Другим недостатком использования NAT является то, что сквозная адресация теряется. Многие интернет-протоколы и приложения зависят от сквозной адресации от источника до места назначения. Некоторые приложения не работают с NAT. Приложения, которые используют физические адреса, а не квалифицированное доменное имя, не доходят до адресатов, которые транслируются через NAT-маршрутизатор. Иногда эту проблему можно избежать, реализуя статические сопоставления NAT;
* Также теряется сквозная трассировка IPv4. Сложнее трассировать пакеты, которые подвергаются многочисленным изменениям адресов пакетов в течение нескольких NAT-переходов, что затрудняет поиск и устранение неполадок;
* Использование NAT также затрудняет протоколы туннелирования, такие как IPsec, поскольку NAT изменяет значения в заголовках, которые мешают проверкам целостности, выполняемым IPsec и другими протоколами туннелирования;
* Службы, требующие инициирования TCP-соединений из внешней сети, или stateless протоколы, например, использующие UDP, могут быть нарушены. Если маршрутизатор NAT не настроен для поддержки таких протоколов, входящие пакеты не могут достичь своего адресата;

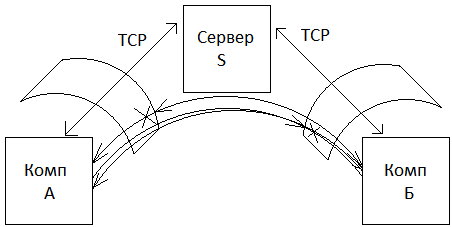
22. Преодоление сетевых экранов при взаимодействии точка-точка по протоколу UDP (UDP Hole Punching).

Если по сети идет широкий поток данных, то соединение через сервер нежелательно (крайне неэффективно и небезопасно), лучше соединение напрямую.

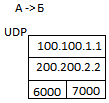


Компьютер А устанавливает соединение с S, при этом сообщая свое имя, свой IP и номер порта, который он использует для связи с собой (выбирается случайно). Сервер возвращает клиенту список его контактов и их IP и порты.

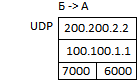
Таким образом, устанавливается и поддерживается ТСР соединение с сервером для аутентификации и получения списка контактов.



Когда Б хочет позвонить А, он уведомляет об этом сервер S. Сервер S уведомляет об этом клиент А по установленному ТСР соединению. После этого А начинает слать UDP пакеты клиенту Б (делает это в цикле):

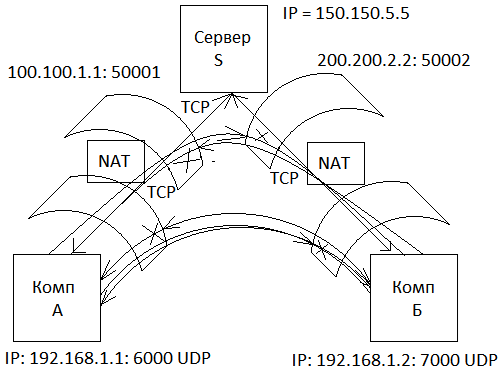


В это же время клиент Б циклически шлет пакеты клиенту А:



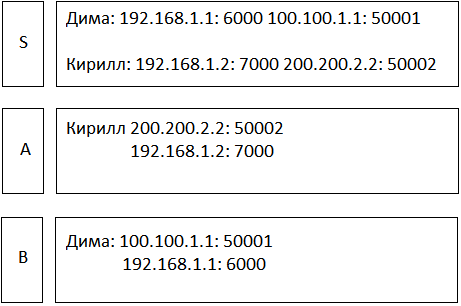
При этом сетевой экран Б не пропускает пакеты от А до тех пор, пока в нем не проделает дыру исходящий пакет от Б. То же самое в отношении А. Но, поскольку пакеты шлются в цикле, то в итоге в обоих экранах начнут возникать дыры и в итоге между А и Б образуется UDP-канал (DUDPHolePunching).

23. Преодоление сетевых экранов при взаимодействии точка-точка по протоколу UDP (UDP Hole Punching) в условиях, когда взаимодействующие узлы используют технологию NAT для выхода в Интернет.

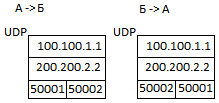


А и Б могут находится как в одной так и в разных локальных сетях.

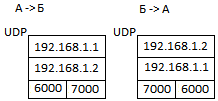
+А устанавливает соединение с S, сообщая при при этом свой локальный IP. S запоминает как этот IP так и IP с которого физически был отправлен пакет.



Далее А и Б начнут слать друг другу пакеты на физические (NAT-овские адреса):



Одновременно (параллельно отправляя на физические адреса) А и Б шлют друг другу пакты на локальные адреса в надежде, что они находятся в одной локальной сети.



Если такое соединение устанавливается, то попытки соединится по внешним адресам прекращаются и соединение работает по локальной сети на огромной скорости.

24. Технология виртуальных сетей VPN. Основные задачи. Способы реализации. Псевдо-каналы. Туннелирование.

***VPN***

**Понятие и классификация *VPN* сетей, их построение**

**Что такое *VPN***

***VPN*** (англ. *Virtual Private Network* - виртуальная частная сеть) - логическая сеть, создаваемая поверх другой сети, например *Internet*. Несмотря на то, что коммуникации осуществляются по публичным сетям с использованием небезопасных протоколов, за счёт шифрования создаются закрытые от посторонних каналы обмена информацией. *VPN* позволяет объединить, например, несколько офисов организации в единую сеть с использованием для связи между ними неподконтрольных каналов.

По своей сути *VPN* обладает многими свойствами выделенной линии, однако развертывается она в пределах общедоступной сети, например Интернета.

Реализация виртуальной частной сети на практике выглядит следующим образом. В локальной вычислительной сети офиса фирмы устанавливается сервер *VPN*. Удаленный пользователь (или маршрутизатор, если осуществляется соединение двух офисов) с использованием клиентского программного обеспечения *VPN* инициирует процедуру соединения с сервером. Происходит аутентификация пользователя - первая фаза установления *VPN*-соединения. В случае подтверждения полномочий наступает вторая фаза - между клиентом и сервером выполняется согласование деталей обеспечения безопасности соединения. После этого организуется *VPN*-соединение, обеспечивающее обмен информацией между клиентом и сервером в форме, когда каждый пакет с данными проходит через процедуры шифрования/дешифрования и проверки целостности - аутентификации данных.

**Классификация *VPN* сетей**

Классифицировать *VPN* решения можно по нескольким основным параметрам:

1. По типу используемой среды:
   * Защищённые *VPN* сети. Наиболее распространённый вариант приватных частных сетей. C его помощью возможно создать надежную и защищенную подсеть на основе ненадёжной сети, как правило, Интернета. Примером защищённых *VPN* являются: *IPSec*, *OpenVPN* и *PPTP*.
   * Доверительные *VPN* сети. Используются в случаях, когда передающую среду можно считать надёжной и необходимо решить лишь задачу создания виртуальной подсети в рамках большей сети. Вопросы обеспечения безопасности становятся неактуальными. Примерами подобных *VPN* решении являются: *MPLS* и *L2TP*. Корректнее сказать, что эти протоколы перекладывают задачу обеспечения безопасности на другие, например *L2TP*, как правило, используется в паре с *IPSec*.
2. По способу реализации:
   * *VPN* сети в виде специального программно-аппаратного обеспечения. Реализация **VPN** сети осуществляется при помощи специального комплекса программно-аппаратных средств. Такая реализация обеспечивает высокую производительность и, как правило, высокую степень защищённости.
   * *VPN* сети в виде программного решения. Используют персональный компьютер со специальным программным обеспечением, обеспечивающим функциональность *VPN*.
   * *VPN* сети с интегрированным решением. Функциональность *VPN* обеспечивает комплекс, решающий также задачи фильтрации сетевого трафика, организации сетевого экрана и обеспечения качества обслуживания.
3. По назначению:
   * *Intranet* *VPN*. Используют для объединения в единую защищённую сеть нескольких распределённых филиалов одной организации, обменивающихся данными по открытым каналам связи.
   * *Remote Access VPN*. Используют для создания защищённого канала между сегментом корпоративной сети (центральным офисом или филиалом) и одиночным пользователем, который, работая дома, подключается к корпоративным ресурсам с домашнего компьютера или, находясь в командировке, подключается к корпоративным ресурсам при помощи ноутбука.
   * *Extranet VPN*. Используют для сетей, к которым подключаются «внешние» пользователи (например, заказчики или клиенты). Уровень доверия к ним намного ниже, чем к сотрудникам компании, поэтому требуется обеспечение специальных «рубежей» защиты, предотвращающих или ограничивающих доступ последних к особо ценной, конфиденциальной информации.
4. По типу протокола:
   * Существуют реализации виртуальных частных сетей под *TCP/IP*, *IPX* и *AppleTalk*. Но на сегодняшний день наблюдается тенденция к всеобщему переходу на протокол *TCP/IP*, и абсолютное большинство *VPN* решений поддерживает именно его.
5. По уровню сетевого протокола:
   * По уровню сетевого протокола на основе сопоставления с уровнями эталонной сетевой модели *ISO/OSI*.

**Построение VPN**

Существуют различные варианты построения *VPN*. При выборе решения требуется учитывать факторы производительности средств построения *VPN*. Например, если маршрутизатор и так работает на пределе мощности своего процессора, то добавление туннелей *VPN* и применение шифрования / дешифрования информации могут остановить работу всей сети из-за того, что этот маршрутизатор не будет справляться с простым трафиком, не говоря уже о *VPN*. Опыт показывает, что для построения *VPN* лучше всего использовать специализированное оборудование, однако если имеется ограничение в средствах, то можно обратить внимание на чисто программное решение. Рассмотрим некоторые варианты построения *VPN*.

* *VPN* на базе брандмауэров. Брандмауэры большинства производителей поддерживают туннелирование и шифрование данных. Все подобные продукты основаны на том, что трафик, проходящий через брандмауэр шифруется. К программному обеспечению собственно брандмауэра добавляется модуль шифрования. Недостатком этого метода можно назвать зависимость производительности от аппаратного обеспечения, на котором работает брандмауэр. При использовании брандмауэров на базе ПК надо помнить, что подобное решение можно применять только для небольших сетей с небольшим объемом передаваемой информации.
* *VPN* на базе маршрутизаторов. Другим способом построения *VPN* является применение для создания защищенных каналов маршрутизаторов. Так как вся информация, исходящая из локальной сети, проходит через маршрутизатор, то целесообразно возложить на этот маршрутизатор и задачи шифрования. Примером оборудования для построения *VPN* на маршрутизаторах является оборудование компании *Cisco Systems*. Начиная с версии программного обеспечения *IOS 11.3*, маршрутизаторы *Cisco* поддерживают протоколы *L2TP* и *IPSec*. Помимо простого шифрования проходящей информации Cisco поддерживает и другие функции *VPN*, такие как идентификация при установлении туннельного соединения и обмен ключами. Для повышения производительности маршрутизатора может быть использован дополнительный модуль шифрования *ESA*.
* *VPN* на базе программного обеспечения. Следующим подходом к построению *VPN* являются чисто программные решения. При реализации такого решения используется специализированное программное обеспечение, которое работает на выделенном компьютере, и в большинстве случаев играет роль proxy-сервера. Компьютер с таким программным обеспечением может быть расположен за брандмауэром.
* *VPN* на базе сетевой ОС.
* *VPN* на базе аппаратных средств. Вариант построения *VPN* на специальных устройствах может быть использован в сетях, требующих высокой производительности.

**Основные составляющие и протоколы *VPN* сетей**

Виртуальная частная сеть базируется на реализации трех составляющих:

· Туннелирование; · Шифрование;

· Аутентификация.

Туннелирование обеспечивает передачу данных между двумя точками - окончаниями туннеля - таким образом, что для источника и приемника данных оказывается скрытой вся сетевая инфраструктура, лежащая между ними.

Транспортная среда туннеля, как паром, подхватывает пакеты используемого сетевого протокола у входа в туннель и без изменений доставляет их к выходу. Построения туннеля достаточно для того, чтобы соединить два сетевых узла так, что с точки зрения работающего на них программного обеспечения они выглядят подключенными к одной (локальной) сети. Однако нельзя забывать, что на самом деле «паром» с данными проходит через множество промежуточных узлов (маршрутизаторов) открытой публичной сети.

Такое положение дел таит в себе две проблемы. Первая заключается в том, что передаваемая через туннель информация может быть перехвачена злоумышленниками. Если она конфиденциальна (номера банковских карточек, финансовые отчеты, сведения личного характера), то вполне реальна угроза ее компрометации, что уже само по себе неприятно. Хуже того, злоумышленники имеют возможность модифицировать передаваемые через туннель данные так, что получатель не сможет проверить их достоверность. Последствия могут быть самыми плачевными. Учитывая сказанное, мы приходим к выводу, что туннель в чистом виде пригоден разве что для некоторых типов сетевых компьютерных игр и не может претендовать на более серьезное применение. Обе проблемы решаются современными средствами криптографической защиты информации, в частности применяются различные методы аутентификации и шифрования.

Чтобы воспрепятствовать внесению несанкционированных изменений в пакет с данными на пути его следования по туннелю, используется метод электронной цифровой подписи. Суть метода состоит в том, что каждый передаваемый пакет снабжается дополнительным блоком информации, который вырабатывается в соответствии с асимметричным криптографическим алгоритмом и уникален для содержимого пакета и секретного ключа ЭЦП отправителя. Этот блок информации является ЭЦП пакета и позволяет выполнить аутентификацию данных получателем, которому известен открытый ключ ЭЦП отправителя. Защита передаваемых через туннель данных от несанкционированного просмотра достигается путем использования сильных алгоритмов шифрования.

С помощью туннелирования пакеты данных транслируются через общедоступную сеть как по обычному двухточечному соединению. Между каждой парой «отправитель-получатель данных» устанавливается своеобразный туннель - безопасное логическое соединение, позволяющее инкапсулировать данные одного протокола в пакеты другого. Основными компонентами туннеля являются:

* инициатор;
* маршрутизируемая сеть;
* туннельный коммутатор;
* один или несколько туннельных терминаторов.

Сам по себе принцип работы *VPN* не противоречит основным сетевым технологиям и протоколам. Например, при установлении соединения удаленного доступа клиент посылает серверу поток пакетов стандартного протокола *PPP*. В случае организации виртуальных выделенных линий между локальными сетями их маршрутизаторы также обмениваются пакетами *PPP*. Тем не менее, принципиально новым моментом является пересылка пакетов через безопасный туннель, организованный в пределах общедоступной сети.

Туннелирование позволяет организовать передачу пакетов одного протокола в логической среде, использующей другой протокол. В результате появляется возможность решить проблемы взаимодействия нескольких разнотипных сетей, начиная с необходимости обеспечения целостности и конфиденциальности передаваемых данных и заканчивая преодолением несоответствий внешних протоколов или схем адресации.

Существующая сетевая инфраструктура корпорации может быть подготовлена к использованию *VPN* как с помощью программного, так и с помощью аппаратного обеспечения. Организацию виртуальной частной сети можно сравнить с прокладкой кабеля через глобальную сеть. Как правило, непосредственное соединение между удаленным пользователем и оконечным устройством туннеля устанавливается по протоколу *PPP*.

Наиболее распространенный метод создания туннелей *VPN* - инкапсуляция сетевых протоколов (*IP*, *IPX*, *AppleTalk* и т.д.) в *PPP* и последующая инкапсуляция образованных пакетов в протокол туннелирования. Обычно в качестве последнего выступает *IP* или (гораздо реже) *ATM* и *Frame Relay*. Такой подход называется туннелированием второго уровня, поскольку «пассажиром» здесь является протокол именно второго уровня.

Альтернативный подход - инкапсуляция пакетов сетевого протокола непосредственно в протокол туннелирования (например, VTP) называется туннелированием третьего уровня.

Независимо от того, какие протоколы используются или какие цели преследуются при организации туннеля, основная методика остается практически неизменной. Обычно один протокол используется для установления соединения с удаленным узлом, а другой - для инкапсуляции данных и служебной информации с целью передачи через туннель.

Сети *VPN* строятся с использованием протоколов туннелирования данных через сеть связи общего пользования Интернет, причем протоколы туннелирования обеспечивают шифрование данных и осуществляют их сквозную передачу между пользователями. Как правило, на сегодняшний день для построения сетей *VPN* используются протоколы следующих уровней:

* Канальный уровень
* Сетевой уровень • Транспортный уровень.

**Канальный уровень**

На канальном уровне могут использоваться протоколы туннелирования данных *L2TP* и *PPTP*, которые используют авторизацию и аутентификацию.

***PPTP*.**

В настоящее время наиболее распространенным протоколом VPN является протокол двухточечной туннельной связи или Point-to-Point Tunnelling Protocol - PPTP. Разработан он компаниями 3Com и Microsoft с целью предоставления безопасного удаленного доступа к корпоративным сетям через Интернет. PPTP использует существующие открытые стандарты TCP/IP и во многом полагается на устаревший протокол двухточечной связи РРР. На практике РРР так и остается коммуникационным протоколом сеанса соединения РРТР. РРТР создает туннель через сеть к NT-серверу получателя и передает по нему РРР-пакеты удаленного пользователя. Сервер и рабочая станция используют виртуальную частную сеть и не обращают внимания на то, насколько безопасной или доступной является глобальная сеть между ними. Завершение сеанса соединения по инициативе сервера, в отличие от специализированных серверов удаленного доступа, позволяет администраторам локальной сети не пропускать удаленных пользователей за пределы системы безопасности Windows Server.

Хотя компетенция протокола РРТР распространяется только на устройства, работающие под управлением Windows, он предоставляет компаниям возможность взаимодействовать с существующими сетевыми инфраструктурами и не наносить вред собственной системе безопасности. Таким образом, удаленный пользователь может подключиться к Интернету с помощью местного провайдера по аналоговой телефонной линии или каналу ISDN и установить соединение с сервером NT. При этом компании не приходится тратить большие суммы на организацию и обслуживание пула модемов, предоставляющего услуги удаленного доступа.

Далее рассматривается работа РРТР. PPTP инкапсулирует пакеты IP для передачи по IP-сети. Клиенты PPTP используют порт назначения для создания управляющего туннелем соединения. Этот процесс происходит на транспортном уровне модели OSI. После создания туннеля компьютер-клиент и сервер начинают обмен служебными пакетами. В дополнение к управляющему соединению PPTP, обеспечивающему работоспособность канала, создается соединение для пересылки по туннелю данных. Инкапсуляция данных перед пересылкой через туннель происходит несколько иначе, чем при обычной передаче. Инкапсуляция данных перед отправкой в туннель включает два этапа:

1. Сначала создается информационная часть PPP. Данные проходят сверху вниз, от прикладного уровня OSI до канального.
2. Затем полученные данные отправляются вверх по модели OSI и инкапсулируются протоколами верхних уровней.

Таким образом, во время второго прохода данные достигают транспортного уровня. Однако информация не может быть отправлена по назначению, так как за это отвечает канальный уровень OSI. Поэтому PPTP шифрует поле полезной нагрузки пакета и берет на себя функции второго уровня, обычно принадлежащие PPP, т.е. добавляет к PPTP-пакету PPP-заголовок и окончание. На этом создание кадра канального уровня заканчивается.

Далее, PPTP инкапсулирует PPP-кадр в пакет Generic Routing Encapsulation (GRE), который принадлежит сетевому уровню. GRE инкапсулирует протоколы сетевого уровня, например IPX, AppleTalk, DECnet, чтобы обеспечить возможность их передачи по IP-сетям. Однако GRE не имеет возможности устанавливать сессии и обеспечивать защиту данных от злоумышленников. Для этого используется способность PPTP создавать соединение для управления туннелем. Применение GRE в качестве метода инкапсуляции ограничивает поле действия PPTP только сетями IP.

После того как кадр PPP был инкапсулирован в кадр с заголовком GRE, выполняется инкапсуляция в кадр с IP-заголовком. IP-заголовок содержит адреса отправителя и получателя пакета. В заключение PPTP добавляет PPP заголовок и окончание.

Система-отправитель посылает данные через туннель. Система-получатель удаляет все служебные заголовки, оставляя только данные PPP.

**L2TP**

L2TP появился в результате объединения протоколов PPTP и L2F (Layer 2 Forwarding). PPTP позволяет передавать через туннель пакеты PPP, а L2F-пакеты SLIP и PPP. Во избежание путаницы и проблем взаимодействия систем на рынке телекоммуникаций, комитет Internet Engineering Task Force

(IETF) рекомендовал компании Cisco Systems объединить PPTP и L2F. По общему мнению, протокол L2TP вобрал в себя лучшие черты PPTP и L2F. Главное достоинство L2TP в том, что этот протокол позволяет создавать туннель не только в сетях IP, но и в таких, как ATM, X.25 и Frame Relay.

L2TP применяет в качестве транспорта протокол UDP и использует одинаковый формат сообщений как для управления туннелем, так и для пересылки данных. L2TP в реализации Microsoft использует в качестве контрольных сообщений пакеты UDP, содержащие шифрованные пакеты PPP. Надежность доставки гарантирует контроль последовательности пакетов.

Функциональные возможности PPTP и L2TP различны. L2TP может использоваться не только в IP-сетях, служебные сообщения для создания туннеля и пересылки по нему данных используют одинаковый формат и протоколы. PPTP может применяться только в IP-сетях, и ему необходимо отдельное соединение TCP для создания и использования туннеля. L2TP поверх IPSec предлагает больше уровней безопасности, чем PPTP, и может гарантировать почти 100-процентную безопасность важных для организации данных. Особенности L2TP делают его очень перспективным протоколом для построения виртуальных сетей.

Протоколы L2TP и PPTP отличаются от протоколов туннелирования третьего уровня рядом особенностей:

1. Предоставление корпорациям возможности самостоятельно выбирать способ аутентификации пользователей и проверки их полномочий - на собственной «территории» или у провайдера Интернет-услуг. Обрабатывая туннелированные пакеты PPP, серверы корпоративной сети получают всю информацию, необходимую для идентификации пользователей.
2. Поддержка коммутации туннелей - завершения одного туннеля и инициирования другого к одному из множества потенциальных терминаторов. Коммутация туннелей позволяет, как бы продлить PPP - соединение до необходимой конечной точки.
3. Предоставление системным администраторам корпоративной сети возможности реализации стратегий назначения пользователям прав доступа непосредственно на брандмауэре и внутренних серверах. Поскольку терминаторы туннеля получают пакеты PPP со сведениями о пользователях, они в состоянии применять сформулированные администраторами стратегии безопасности к трафику отдельных пользователей. (Туннелирование третьего уровня не позволяет различать поступающие от провайдера пакеты, поэтому фильтры стратегии безопасности приходится применять на конечных рабочих станциях и сетевых устройствах.) Кроме того, в случае использования туннельного коммутатора появляется возможность организовать «продолжение» туннеля второго уровня для непосредственной трансляции трафика отдельных пользователей к соответствующим внутренним серверам. На такие серверы может быть возложена задача дополнительной фильтрации пакетов. **· MPLS**

Также на канальном уровне для организации туннелей может использоваться технология MPLS (От английского Multiprotocol Label Switching - мультипротокольная коммутация по меткам - механизм передачи данных, который эмулирует различные свойства сетей с коммутацией каналов поверх сетей с коммутацией пакетов). MPLS работает на уровне, который можно было бы расположить между канальным и третьим сетевым уровнями модели OSI, и поэтому его обычно называют протоколом канально-сетевого уровня. Он был разработан с целью обеспечения универсальной службы передачи данных как для клиентов сетей с коммутацией каналов, так и сетей с коммутацией пакетов. С помощью MPLS можно передавать трафик самой разной природы, такой как IP-пакеты, ATM, SONET и кадры Ethernet.

Решения по организации VPN на канальном уровне имеют достаточно ограниченную область действия, как правило, в рамках домена провайдера.

**Сетевой уровень**

Сетевой уровень (уровень IP). Используется протокол IPSec реализующий шифрование и конфедициальность данных, а также аутентификацию абонентов. Применение протокола IPSec позволяет реализовать полнофункциональный доступ эквивалентный физическому подключению к корпоративной сети. Для установления VPN каждый из участников должен сконфигурировать определенные параметры IPSec, т.е. каждый клиент должен иметь программное обеспечение реализующее IPSec.

**IPSec**

Естественно, никакая компания не хотела бы открыто передавать в Интернет финансовую или другую конфиденциальную информацию. Каналы VPN защищены мощными алгоритмами шифрования, заложенными в стандарты протокола безопасности IРsec. IPSec или Internet Protocol Security - стандарт, выбранный международным сообществом, группой IETF - Internet Engineering Task Force, создает основы безопасности для Интернет-протокола (IP/ Протокол IPSec обеспечивает защиту на сетевом уровне и требует поддержки стандарта IPSec только от общающихся между собой устройств по обе стороны соединения. Все остальные устройства, расположенные между ними, просто обеспечивают трафик IP-пакетов.

Способ взаимодействия лиц, использующих технологию IPSec, принято определять термином «защищенная ассоциация» - Security Association (SA). Защищенная ассоциация функционирует на основе соглашения, заключенного сторонами, которые пользуются средствами IPSec для защиты передаваемой друг другу информации. Это соглашение регулирует несколько параметров: IP-адреса отправителя и получателя, криптографический алгоритм, порядок обмена ключами, размеры ключей, срок службы ключей, алгоритм аутентификации.

IPSec - это согласованный набор открытых стандартов, имеющий ядро, которое может быть достаточно просто дополнено новыми функциями и протоколами. Ядро IPSec составляют три протокола:

· **АН** или Authentication Header - заголовок аутентификации - гарантирует целостность и аутентичность данных. Основное назначение протокола АН - он позволяет приемной стороне убедиться, что:

• пакет был отправлен стороной, с которой установлена безопасная ассоциация; • содержимое пакета не было искажено в процессе его передачи по сети; • пакет не является дубликатом уже полученного пакета.

Две первые функции обязательны для протокола АН, а последняя выбирается при установлении ассоциации по желанию. Для выполнения этих функций протокол АН использует специальный заголовок. Его структура рассматривается по следующей схеме:

1. В поле следующего заголовка (next header) указывается код протокола более высокого уровня, то есть протокола, сообщение которого размещено в поле данных IP-пакета.
2. В поле длины полезной нагрузки (payload length) содержится длина заголовка АН.
3. Индекс параметров безопасности (Security Parameters Index, SPI) используется для связи пакета с предусмотренной для него безопасной ассоциацией.
4. Поле порядкового номера (Sequence Number, SN) указывает на порядковый номер пакета и применяется для защиты от его ложного воспроизведения (когда третья сторона пытается повторно использовать перехваченные защищенные пакеты, отправленные реально аутентифицированным отправителем).
5. Поле данных аутентификации (authentication data), которое содержит так называемое значение проверки целостности (Integrity Check Value, ICV), используется для аутентификации и проверки целостности пакета. Это значение, называемое также дайджестом, вычисляется с помощью одной из двух обязательно поддерживаемых протоколом АН вычислительно необратимых функций MD5 или SAH-1, но может использоваться и любая другая функция.

· **ESP или Encapsulating Security Payload** - инкапсуляция зашифрованных данных - шифрует передаваемые данные, обеспечивая конфиденциальность, может также поддерживать аутентификацию и целостность данных;

Протокол ESP решает две группы задач.

1. К первой относятся задачи, аналогичные задачам протокола АН, - это обеспечение аутентификации и целостности данных на основе дайджеста,
2. Ко второй - защита передаваемых данных путем их шифрования от несанкционированного просмотра.

Заголовок делится на две части, разделяемые полем данных.

1. Первая часть, называемая собственно заголовком ESP, образуется двумя полями (SPI и SN), назначение которых аналогично одноименным полям протокола АН, и размещается перед полем данных.
2. Остальные служебные поля протокола ESP, называемые концевиком ESP, расположены в конце пакета.

Два поля концевика - следующего заголовка и данных аутентификации - аналогичны полям заголовка АН. Поле данных аутентификации отсутствует, если при установлении безопасной ассоциации принято решение не использовать возможностей протокола ESP по обеспечению целостности. Помимо этих полей концевик содержит два дополнительных поля заполнителя и длины заполнителя.

Протоколы AH и ESP могут защищать данные в двух режимах:

1. в транспортном - передача ведется с оригинальными IP-заголовками;
2. в туннельном - исходный пакет помещается в новый IP-пакет и передача ведется с новыми заголовками.

Применение того или иного режима зависит от требований, предъявляемых к защите данных, а также от роли, которую играет в сети узел, завершающий защищенный канал. Так, узел может быть хостом (конечным узлом) или шлюзом (промежуточным узлом).

Соответственно, имеются три схемы применения протокола IPSec:

1. хост-хост;
2. шлюз-шлюз;
3. хост-шлюз.

Возможности протоколов АН и ESP частично перекрываются: протокол АН отвечает только за обеспечение целостности и аутентификации данных, протокол ESP может шифровать данные и, кроме того, выполнять функции протокола АН (в урезанном виде). ESP может поддерживать функции шифрования и аутентификации / целостности в любых комбинациях, то есть либо всю группу функций, либо только аутентификацию / целостность, либо только шифрование.

· IKE или Internet Key Exchange - обмен ключами Интернета - решает вспомогательную задачу автоматического предоставления конечным точкам защищенного канала секретных ключей, необходимых для работы протоколов аутентификации и шифрования данных.

**Транспортный уровень**

На транспортном уровне используется протокол SSL/TLS или Secure Socket Layer/Transport Layer Security, реализующий шифрование и аутентификацию между транспортными уровнями приемника и передатчика. SSL/TLS может применяться для защиты трафика TCP, не может применяться для защиты трафика UDP. Для функционирования VPN на основе SSL/TLS нет необходимости в реализации специального программного обеспечения так как каждый браузер и почтовый клиент оснащены этими протоколами. В силу того, что SSL/TLS реализуется на транспортном уровне, защищенное соединение устанавливается «из-конца-в-конец».

TLS-протокол основан на Netscape SSL-протоколе версии 3.0 и состоит из двух частей - TLS Record Protocol и TLS Handshake Protocol. Различие между SSL 3.0 и TLS 1.0 незначительные.

SSL/TLS включает в себя три основных фазы:

1. Диалог между сторонами, целью которого является выбор алгоритма шифрования;
2. Обмен ключами на основе криптосистем с открытым ключом или аутентификация на основе сертификатов;
3. Передача данных, шифруемых при помощи симметричных алгоритмов шифрования.

**Еще раз об аутентификации и шифровании**

Обеспечение безопасности является основной функцией VPN. Все данные от компьютеровклиентов проходят через *Internet* к VPN-серверу. Такой сервер может находиться на большом расстоянии от клиентского компьютера, и данные на пути к сети организации проходят через оборудование множества провайдеров. Как убедиться, что данные не были прочитаны или изменены? Для этого применяются различные методы аутентификации и шифрования.

Для аутентификации пользователей PPTP может задействовать любой из протоколов, применяемых для PPP

* EAP или Extensible Authentication Protocol;
* MSCHAP или Microsoft Challenge Handshake Authentication Protocol (версии 1 и 2);
* CHAP или Challenge Handshake Authentication Protocol; • SPAP или Shiva Password Authentication Protocol; • PAP или Password Authentication Protocol.

Лучшими считаются протоколы MSCHAP версии 2 и Transport Layer Security (EAP-TLS), поскольку они обеспечивают взаимную аутентификацию, т.е. VPN-сервер и клиент идентифицируют друг друга. Во всех остальных протоколах только сервер проводит аутентификацию клиентов.

Шифрование с помощью PPTP гарантирует, что никто не сможет получить доступ к данным при пересылке через Internet. В настоящее время поддерживаются два метода шифрования:

* Протокол шифрования MPPE или Microsoft Point-to-Point Encryption совместим только с MSCHAP (версии 1 и 2);
* EAP-TLS и умеет автоматически выбирать длину ключа шифрования при согласовании параметров между клиентом и сервером.

MPPE поддерживает работу с ключами длиной 40, 56 или 128 бит. Старые операционные системы Windows поддерживают шифрование с длиной ключа только 40 бит, поэтому в смешанной среде Windows следует выбирать минимальную длину ключа.

PPTP изменяет значение ключа шифрации после каждого принятого пакета. Протокол MMPE разрабатывался для каналов связи точка-точка, в которых пакеты передаются последовательно, и потеря данных очень мала. В этой ситуации значение ключа для очередного пакета зависит от результатов дешифрации предыдущего пакета. При построении виртуальных сетей через сети общего доступа эти условия соблюдать невозможно, так как пакеты данных часто приходят к получателю не в той последовательности, в какой были отправлены. Поэтому PPTP использует для изменения ключа шифрования порядковые номера пакетов. Это позволяет выполнять дешифрацию независимо от предыдущих принятых пакетов.

Хотя PPTP обеспечивает достаточную степень безопасности, но все же L2TP поверх IPSec надежнее. L2TP поверх IPSec обеспечивает аутентификацию на уровнях «пользователь» и 'компьютер', а также выполняет аутентификацию и шифрование данных.

Аутентификация осуществляется либо отрытым тестом (clear text password), либо по схеме запрос / отклик (challenge/response). С прямым текстом все ясно. Клиент посылает серверу пароль. Сервер сравнивает это с эталоном и либо запрещает доступ, либо говорит «добро пожаловать».

Открытая аутентификация практически не встречается.

Схема запрос / отклик намного более продвинута. В общем виде она выглядит так:

* клиент посылает серверу запрос (request) на аутентификацию;
* сервер возвращает случайный отклик (challenge);
* клиент снимает со своего пароля хеш (хешем называется результат хеш-функции, которая преобразовывает входной массив данных произвольной длины в выходную битовую строку фиксированной длины), шифрует им отклик и передает его серверу;
* то же самое проделывает и сервер, сравнивая полученный результат с ответом клиента;
* если зашифрованный отклик совпадает, аутентификация считается успешной;

На первом этапе аутентификации клиентов и серверов VPN, L2TP поверх IPSec использует локальные сертификаты, полученные от службы сертификации. Клиент и сервер обмениваются сертификатами и создают защищенное соединение ESP SA (security association). После того как L2TP (поверх IPSec) завершает процесс аутентификации компьютера, выполняется аутентификация на уровне пользователя. Для аутентификации можно задействовать любой протокол, даже PAP, передающий имя пользователя и пароль в открытом виде. Это вполне безопасно, так как L2TP поверх IPSec шифрует всю сессию. Однако проведение аутентификации пользователя при помощи MSCHAP, применяющего различные ключи шифрования для аутентификации компьютера и пользователя, может усилить защиту.

Таким образом, связка «туннелирование + аутентификация + шифрование» позволяет передавать данные между двумя точками через сеть общего пользования, моделируя работу частной (локальной) сети. Иными словами, рассмотренные средства позволяют построить виртуальную частную сеть.