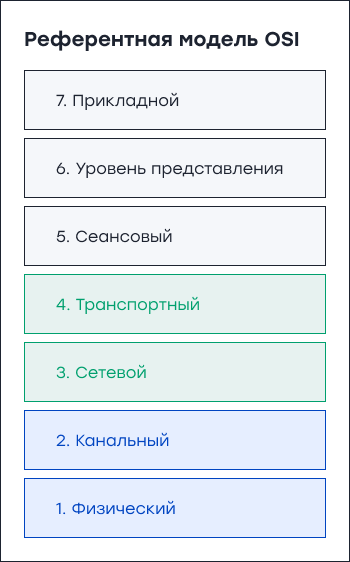
**Занятие 3**

**OSI**

Как вы поняли из истории, это набор правил, который описывает процесс взаимодействия устройств по сети. OSI выступает первой стандартной моделью в области сетевых коммуникаций.



По модели процесс передачи данных по сети происходит постепенно от одного уровня к другому. На каждом из них используются информация с прошлого уровня и определенные протоколы. Главными героями здесь выступают устройства отправителя и получателя, а также сами передаваемые данные. И как раз процесс обмена информации между устройствами определяет модель OSI.

На физическом уровне информация предстает в виде битов, а на прикладном она отражается в более привычном для нас виде, в виде данных. Существует два процесса перехода от первого уровня к седьмому и наоборот. Первый – это инкапсуляция, когда данные отправляются с устройства и переводятся в биты. Второй – декапсуляция, обратный переход, когда биты трансформируются в данные.

Разбираемся, что конкретно делают уровни, и что же там происходит. Смотрим на модель снизу вверх.

**Уровень 1: Физический**

Начнем (кто бы удивился) с уровня 1. Здесь происходит обмен оптическими, электрическими или радиосигналами между устройствами отправителя и получателя.

На этом уровне железо не распознает данные в классическом для нас виде (картинки, текст, видео), но оно понимает биты (единицы и нули) и работает только с сигналами. Таким оборудованием выступают концентраторы, медиаконвертеры или репитеры. Здесь информация или биты передаются либо по проводам, кабелям, либо без них, например через Bluetooth, Wi-Fi.

Когда возникает проблема с сетью, многие специалисты сразу же обращаются к физическому уровню, чтобы проверить, например, не отключен ли сетевой кабель от устройства.

**Уровень 2: Канальный**

Мы прошли первый уровень. Что же дальше? Если в локальной сети находится более двух устройств, то необходимо определить, куда конкретно направлять информацию. Этим занимается как раз канальный уровень, принимающий на себя важную роль адресации.

Второй уровень принимает биты и трансформирует их в кадры (фреймы). Здесь существуют MAC-адреса (Media Access Control), которые необходимы для идентификации устройств. На втором уровне происходит еще проверка на ошибки, и исправление информации, а также управление ее передачей. Этим занимается LLC (Logical Link Control).

На канальном уровне работают уже более умные железки – коммутаторы. Их задачей является передача кадров от одного устройства другому, используя MAC-адреса.

**Уровень 3: Сетевой**

На третьем уровне происходит маршрутизация трафика. Этим занимаются такие устройства, как роутеры или маршрутизаторы.

На сетевом уровне работает протокол ARP (Address Resolution Protocol), который определяет соответствие между логическим адресом сетевого уровня (IP) и физическим адресом устройства (MAC). Здесь пересылаемая информация выступает уже в виде пакетов, состоящих из заголовка и поля данных.

Информация об известных IP и MAC-адресах хранится в виде таблицы (ARP-таблица) с данными, что позволяет устройствам не тратить время на повторную идентификацию.

**Уровень 4: Транспортный**

Четвертый уровень получает пакеты и передает их по сети. Он отвечает за установку соединения, надежность и управление потоком. Блоки данных делятся на отдельные фрагменты, размеры которых зависят от используемого протокола. Главными героями тут выступают 2 протокола TCP (Transmission Control Protocol) и UDP (User Datagram Protocol). В чем их отличие и когда их применять?

При транспортировке данных, наиболее восприимчивых к потерям, например, web-страницы, задействуется протокол TCP с установлением соединения. Он контролирует целостность информации, в данном случае нашей страницы, ибо потеря какого-то контента заставит задуматься пользователя о его полезности. Чтобы сделать передачу более эффективной и быстрой, транспортный уровень разбивает данные на более мелкие сегменты.

UDP-протокол используется с данными, для которых потери не так критичны, например, мультимедиа-трафик. Для них более заметна будет задержка, поэтому UDP обеспечивает связь без установки соединения. Во время передачи данных с помощью протокола UDP, пакеты делятся уже на автономные датаграммы. Они могут доставляться по разным маршрутам и в разной последовательности.

**Уровень 5: Сеансовый**

Уровни с пятого по седьмой уже работают с чистыми данными. И здесь за дело берутся не сетевые инженеры, а разработчики.

Сеансовый уровень, исходя из названия, отвечает за поддержание сеанса или сессии. Он координирует коммуникацию между приложениями и отвечает за установление, поддержание и завершение связи, синхронизацию задач и сам обмен информацией. Примером для пятого уровня можно назвать созвон в Zoom или прямой эфир на YouTube. Во время сессии необходимо обеспечивать синхронизированную передачу аудио и видео для всех участников, а также поддерживать саму связь. За это как раз отвечают протоколы сеансового уровня (RPC, H.245, RTCP).

**Уровень 6: Уровень представления**

Шестой уровень подготавливает информацию для последнего и преобразует (сжимает, кодирует, шифрует) их в понятный язык для пользователя или машины. Например, если вы отправляете картинку, то она сначала приходит в виде битов, а потом трансформируются в JPEG, GIF или другой формат.

**Уровень 7: Прикладной**

Верхний уровень модели OSI – это прикладной. С помощью своих протоколов он отображает данные в понятном конечному пользователю формате. Сюда входят такие технологии, как HTTP, DNS, FTP, SSH и многое другое. Почти каждый человек ежедневно взаимодействует с протоколами прикладного уровня.

**Как это все работает?**

Чтобы информация могла быть передана по сети от устройства к устройству, данные должны пройти семь кругов, а точнее уровней по модели OSI. Информация передается с уровня 7 вниз на уровень 1 от отправителя, а затем передается с уровня 1 на уровень 7 на устройстве получателя.

Примером передачи данных по модели OSI является приложение электронной почты. Когда пользователь отправляет письмо, оно приходит на уровень представления с использованием определенного протокола (SMTP для исходящей электронной почты). Уровень представления сжимает информацию и отправляет сообщение на сеансовый, который открывает сессию для связи между устройством отправителя и исходящим сервером.

Далее вступает в силу транспортный уровень, где сегментируются полученные данные. Затем сетевой уровень разбивает сегменты на пакеты и отправляет их на канальный уровень, где они разбиваются на фреймы. Фреймы переходят на физический уровень, где информация преобразуется в биты и передается через физическую среду, ​​беспроводные соединения или кабели.

Когда сообщение доходит до получателя, происходит обратный процесс, где информация переходит из битовых единиц и нулей в сообщение на почте получателя. Как-то так.

Информация взята из сатьи Habr <https://habr.com/ru/companies/serverspace/articles/689704/>

**TCP/IP**

Протокол передачи данных, который используется в современной сети Ethernet.

**МАС адрес**

Все началось с MAC-адреса. МАС адрес (media access control address) – уникальный идентификатор, назначенный сетевому адаптеру, применяется в сетях стандартов IEEE 802, в основном Ethernet, Wi-Fi и Bluetooth. Официально он называется «идентификатором типа EUI-48». Из названия очевидно, что адрес имеет длину в 48 бит, т.е. 6 байт. Общепринятого стандарта на написание адреса нет (в противоположность IPv4 адресу, где октеты всегда разделяют точками).Обычно он записывается как шесть шестнадцатеричных чисел, разделенных двоеточием: 00:AB:CD:EF:11:22, хотя некоторые производители оборудования предпочитают запись вида 00-AB-CD-EF-11-22 и даже 00ab.cdef.1122. Пример компаний, которым принадлежат определенные части адреса:

|  |  |  |
| --- | --- | --- |
| **Вендор** | **Число блоков/записей** | **Число адресов, млн.** |
| Cisco Systems Inc | 888 | 14208 |
| Apple | 772 | 12352 |
| Samsung | 636 | 10144 |
| Huawei Technologies Co.Ltd | 606 | 9696 |
| Intel Corporation | 375 | 5776 |
| ARRIS Group Inc. | 319 | 5104 |
| Nokia Corporation | 241 | 3856 |
| Private | 232 | 2704 |
| Texas Instruments | 212 | 3392 |
| zte corporation | 198 | 3168 |
| IEEE Registration Authority | 194 | 3072 |
| Hewlett Packard | 149 | 2384 |
| Hon Hai Precision | 136 | 2176 |
| TP-LINK | 134 | 2144 |
| Dell Inc. | 123 | 1968 |
| Juniper Networks | 110 | 1760 |
| Sagemcom Broadband SAS | 97 | 1552 |
| Fiberhome Telecommunication Technologies Co. LTD | 97 | 1552 |
| Xiaomi Communications Co Ltd | 88 | 1408 |
| Guangdong Oppo Mobile Telecommunications Corp.Ltd | 82 | 1312 |

Во время глобализации возникла проблема, что два устройства, произведенные в одном месте и имеющие одинаковые MAC-адреса могут использоваться в разных странах. Чтобы пакет достиг нужного места придется сканировать всю сеть, чтобы найти путь до необходимого хоста. Для решения этой проблемы необходимо географически разделить сеть на подсети. На помощи пришел IP адрес.

**IP адрес**

IP-адрес — уникальный сетевой адрес узла в компьютерной сети, построенной на основе стека протоколов TCP/IP (TCP/IP – это набор интернет-протоколов, о котором мы поговорим в дальнейших статьях). IP-адрес представляет собой серию из 32 двоичных бит (единиц и нулей). Так как человек невосприимчив к большому однородному ряду чисел, такому как этот 11100010101000100010101110011110 (здесь, к слову, 32 бита информации, так как 32 числа в двоичной системе), было решено разделить ряд на четыре 8-битных байта и получилась следующая последовательность: 11100010.10100010.00101011.10011110. Это не сильно облегчило жизнь и было решение перевести данную последовательность в, привычную нам, последовательность из четырёх чисел в десятичной системе, то есть 226.162.43.158. 4 разряда также называются октетами. Данный IP адрес определяется протоколом IPv4. По такой схеме адресации можно создать более 4 миллиардов IP-адресов.  
  
Максимальным возможным числом в любом октете будет 255 (так как в двоичной системе это 8 единиц), а минимальным – 0.  
  
Далее давайте разберёмся с тем, что называется классом IP (именно в этом моменте в лабораторной работе была неточность).  
  
IP-адреса делятся на 5 классов (A, B, C, D, E). A, B и C — это классы коммерческой адресации. D – для многоадресных рассылок, а класс E – для экспериментов.  
  
Класс А: 1.0.0.0 — 126.0.0.0, маска 255.0.0.0  
Класс В: 128.0.0.0 — 191.255.0.0, маска 255.255.0.0  
Класс С: 192.0.0.0 — 223.255.255.0, маска 255.255.255.0  
Класс D: 224.0.0.0 — 239.255.255.255, маска 255.255.255.255  
Класс Е: 240.0.0.0 — 247.255.255.255, маска 255.255.255.255

Но классы не используются. Можно на них забить. Однако из них к нам приходит понятие масок подсети.

**Маски подсети**

У всех IP адресов есть две части сеть и хост.  
Сеть – это та часть IP, которая не меняется во всей сети и все адреса устройств начинаются именно с номера сети.  
Узел – это изменяющаяся часть IP. Каждое устройство имеет свой уникальный адрес в сети, он называется узлом.

Маску принято записывать двумя способами: префиксным и десятичным. Например, маска частной подсети A выглядит в десятичной записи как 255.0.0.0, но не всегда удобно пользоваться десятичной записью при составлении схемы сети. Легче записать маску как префикс, то есть /8.  
  
Так как маска формируется добавлением слева единицы с первого октета и никак иначе, но для распознания маски нам достаточно знать количество выставленных единиц.

Для понимания представим IP адрес и маску в двоичном коде

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP | 0000 0000 | . | 0000 0000 | . | 0000 0000 | . | 0000 0000 |
| маска | 0000 0000 | . | 0000 0000 | . | 0000 0000 | . | 0000 0000 |

Маска указывает на биты IP адреса, которые отведены на адрес сети, остальные биты для адресов хостов. Например, если маска 255.255.0.0, то для IP адреса первые 8 бит указывают на адрес сети, а остальные на хост. В примере IP 192.168.10.3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP | 1100 0000 | . | 1010 1000 | . | 0000 1010 | . | 0000 0011 |
| маска | 1111 1111 | . | 1111 1111 | . | 0000 0000 | . | 0000 0000 |

Для IP 192.168.128.1 и маски 255.255.254.0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0000 0001 |
| маска | 1111 1111 | . | 1111 1111 | . | 1111 1110 | . | 0000 0000 |

Для IP 192.168.128.1 и маски 255.255.255.0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0000 0001 |
| маска | 1111 1111 | . | 1111 1111 | . | 1111 1111 | . | 0000 0000 |

Для IP 192.168.128.1 и маски 255.255.255.128

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0000 0001 |
| маска | 1111 1111 | . | 1111 1111 | . | 1111 1111 | . | 1000 0000 |

Для IP 192.168.128.1 и маски 255.255.255.192

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0000 0001 |
| маска | 1111 1111 | . | 1111 1111 | . | 1111 1111 | . | 1100 0000 |

Теперь разберем подсчет сетей. В основном задача задача заключается в том, чтобы составить маску подсети такую, чтобы можно было составить определенное количество подсетей и и хостов в них. Последние биты отводятся под хост, подсчитав их, мы может определить количество ip адресов в данной сети.

Высчитаем сколько устройств (в IP адресах — узлов) может быть в сети, где у одного компьютера адрес 172.16.13.98 /24.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP | 1010 1100 | . | 0001 0000 | . | 0000 1101 | . | 0110 0010 |
| маска | 1111 1111 | . | 1111 1111 | . | 1111 1111 | . | 0000 0000 |

На адреса уходит 8 бит. Всего может быть 28 = 256 адресов. Но существуют адреса, которые являются специальными.

Адрес сети это самое минимальное число, которое определяет саму сеть. Устройства в сети не могут иметь ip, схожий с адресом сети. Для нашего примера адрес сети 172.16.13.0.

Широко вещательный адрес. Это адрес, позволяющий обратиться сразу ко всем устройствам в подсети. Он также зарезервирован и не может быть присвоен устройством. Для нашей сети это 172.16.13.255.

По итогу получается, что для нашей подсети существует всего 254 IP адреса, которые мы можем присвоить сетевым устройствам. Количество хостов в сети можно посчитать по формуле:

N = 2m - 2

где m – количество бит, отведенных под хосты.

172.16.13.0 – адрес сети  
172.16.13.1 – адрес первого устройства в сети  
172.16.13.254 – адрес последнего устройства в сети  
172.16.13.255 – широковещательный IP адрес  
172.16.14.0 – адрес следующей сети  
  
**Итого** 254 устройства в сети  
  
Теперь вычислим сколько устройств может быть в сети, где у одного компьютера адрес 172.16.13.98 /16.  
172.16.0.0 – адрес сети  
172.16.0.1 – адрес первого устройства в сети  
172.16.255.254 – адрес последнего устройства в сети  
172.16.255.255 – широковещательный IP адрес  
172.17.0.0 – адрес следующей сети  
  
**Итого** 65534 устройства в сети

**Задача**

Выполним стандартную задачу. Дана сеть 192.168.128.0 / 24. Необходимо разбить на 3 подсети. В первой должно быть 120 хостов, во второй 50, в третьей 60.

**Решение**

Смотрим на нашу сеть

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0000 0000 |
| маска | 1111 1111 | . | 1111 1111 | . | 1111 1111 | . | 0000 0000 |

Здесь можно видеть, что всего мы можем поддерживать 254 хоста. Наша задача решаема. Мы имеем 24 маску, из нее мы можем составить 25, 26, 27, 28 ... Для такого разделения необходимо отдавать новые биты в адрес сети.

Начнем с первой подсети. По условию нам необходимо 120 хостов, если мы посмотрим ближайший адрес с маской 25, мы увидим возможность разместить 126 узлов. Берем его для первой подсети

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0000 0000 |
| маска | 1111 1111 | . | 1111 1111 | . | 1111 1111 | . | 1000 0000 |
| Первый | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0000 0001 |
| Последний | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0111 1110 |
| Broadcast | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0111 1111 |

|  |  |
| --- | --- |
| Адрес | 192.168.128.0 |
| Bitmask | 25 |
| Netmask | 255.255.255.128 |
| Wildcard | 0.0.0.127 |
| Network | 192.168.128.0 |
| Broadcast | 192.168.128.127 |
| Hostmin | 192.168.128.1 |
| Hostmax | 192.168.128.126 |
| Hosts | 126 |

Теперь считаем для второй подсети

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0100 0000 |
| маска | 1111 1111 | . | 1111 1111 | . | 1111 1111 | . | 1100 0000 |
| Первый | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0100 0001 |
| Последний | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0111 1110 |
| Broadcast | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 0111 1111 |

|  |  |
| --- | --- |
| Адрес | 192.168.128.64 |
| Bitmask | 26 |
| Netmask | 255.255.255.192 |
| Wildcard | 0.0.0.63 |
| Network | 192.168.128.64 |
| Broadcast | 192.168.128.127 |
| Hostmin | 192.168.128.65 |
| Hostmax | 192.168.128.126 |
| Hosts | 62 |

И для третьей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IP | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 1000 0000 |
| маска | 1111 1111 | . | 1111 1111 | . | 1111 1111 | . | 1100 0000 |
| Первый | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 1000 0001 |
| Последний | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 1011 1110 |
| Broadcast | 1100 0000 | . | 1010 1000 | . | 1000 0000 | . | 1011 1111 |

|  |  |
| --- | --- |
| Адрес | 192.168.128.128 |
| Bitmask | 26 |
| Netmask | 255.255.255.192 |
| Wildcard | 0.0.0.63 |
| Network | 192.168.128.128 |
| Broadcast | 192.168.128.191 |
| Hostmin | 192.168.128.129 |
| Hostmax | 192.168.128.190 |
| Hosts | 62 |

**Виды протоколов**

|  |  |  |
| --- | --- | --- |
| Протокол | Описание | Уровень на OSI |
| TCP | Протокол обмена сообщениями в сети Интернет. Для передачи информации происходит дробление исходного файла на части, которые передаются получателю, а далее собираются обратно. Например, человек запрашивает веб-страницу, далее сервер обрабатывает запрос и высылает в ответ HTML-страницу при помощи протокола HTTP. Он, в свою очередь, запрашивает уровень TCP для установки требуемого соединения и отправки HTML-файла. TCP конвертирует данные в блоки, передавая их на уровень TCP пользователя, где происходит подтверждение передачи. | 4 |
| UDP | Аналог TCP. В отличие от протокола ТСР User Datagram Protocol обеспечивает передачу данных без получения подтверждения от пользователя о результате действия. Благодаря этому достигается большая скорость работы и передачи данных в ущерб надежности и безопасности. | 4 |
| SCTP | Протокол передачи управления потоком. Еще один протокол, который относится к транспортному уровню. SCTP обеспечивает надежную последовательную передачу данных. Поддерживает многоадресное соединение, когда один или оба конечных узла могут состоять из более чем одного IP-адреса. Это обеспечивает прозрачное переключение между резервными сетевыми путями. | 4 |
| RTP | Real-time Transfer Protocol — это протокол, который используется при передаче потокового аудио и видео и применяется при передаче голоса преимущественно в IP-телефонии. RTP применяется в совокупности с протоколом управления RTCP. Когда RTP транслирует медиа, RTCP применяется при анализе статистик QoS (Quality of Service) и обеспечивает синхронизацию разных потоков. RTP отправляется и принимается с помощью четных номеров портов, а RTCP использует нечетные номера. | 4 |
| ICMP | Основная цель ICMP — сообщать об ошибках. Если какие-либо данные не попали по назначению, ICMP генерирует ошибки для обмена с отправляющим устройством. Например, если объем передаваемых данных слишком велик для маршрутизатора, маршрутизатор отбросит пакет и отправит ICMP-сообщение исходному источнику данных. | 3 |
| OSPF | После настройки OSPF будет анализировать соседние маршрутизаторы и собирать все доступные данные о состоянии канала для построения топологической карты всех доступных путей в своей сети. Затем он сохранит информацию в своей базе данных топологии, также известной как База данных состояния канала (LSDB). | 3 |
| FTP | FTP — это клиент-серверный протокол, который использует два канала для передачи данных: командный, управляющий процессом передачи, и транспортный, непосредственно передающий информацию. Для FTP-протокола устройство конечного пользователя называется локальным хостом, а второй компьютер — удаленным хостом, играющим роль сервера. Для работы протокола требуется его правильная настройка со стороны хоста и специальный клиент на локальном хосте. | 7 |
| DNS | Браузеры взаимодействуют между собой через IP-адреса. Люди, пытаясь подключиться к сайту, используют его доменное имя — например, https://selectel.ru/. Domain Name System преобразует домены в IP-адреса, чтобы сделать возможной загрузку интернет-ресурса через браузер. Каждому устройству в сети назначается свой IP-адрес, который используется другими устройствами для подключения к нему, а DNS-сервер позволяет людям не запоминать их. | 7 |
| HTTP(S) | Обычно принцип передачи данных по протоколу HTTP включает в себя компьютер клиента (например, ваш ПК), отправляющий запрос на сервер, который затем возвращает ответ. Каждый HTTP-запрос включает в себя ряд закодированных данных, содержащих различную информацию, в том числе:  версию HTTP,  URL-адрес,  метод HTTP-запроса — указание на ожидание запроса от сервера (например, PUT- и GET-запросы),  заголовок — он передает основную информацию о запросе и содержит пары ключ-значение,  тело запроса (опционально, это любая отправляющаяся информация). | 7 |
| SSH | SSH, или Secure Shell, — это защищенный протокол, который используется как основное средство подключения к серверам. С помощью него при подключении к серверу пользователь входит в уже существующую учетную запись, где выполняются все отправленные команды. Данное соединение реализовано по схеме «клиент-сервер». Для его создания на удаленном устройстве должна быть запущена программа, называемая демоном. Демон выполняет подключение к определенному сетевому порту, проверяет подлинность запросов на подключение и создает соответствующую среду, если пользователь вводит правильные учетные данные. Также со стороны клиента должно быть установлено соответствующее ПО. | 7 |