## Обзор TCP/IP

### 1.1 История TCP/IP

Термин TCP/IP относится к целому семейству протоколов и образован из названий двух из них: Transmission Control Protocol (TCP) и Internet Protocol (IP). Протоколы семейства TCP/IP начали разрабатываться как часть экспериментальной сети ARPAnet, созданной Агентством перспективных исследований Министерства обороны США (United States Defense Advanced Research Projects Agency, DARPA или ARPA). Первоначально сеть ARPAnet использовала адаптированные к её требованиям существующие на тот момент протоколы. Однако все они имели какие-либо недостатки или ограничения. Разработчики новой сети поняли, что использование имеющихся протоколов приведет к существенным проблемам по мере ее расширения.

В 1973 году началась разработка полноценной системы протоколов межсетевого обмена для сети ARPAnet. Самая ранняя её версия, написанная в 1973 году, содержала описание только одного протокола: TCP. Эта аббревиатура означала «Transmission Control Program». Далее эта версия была доработана, и в декабре 1974 года формально документирована в RFC 675 «Specification of Internet Transmission Control Program».

Тестирование и исследование TCP продолжалось несколько лет. В марте 1977 года была документирована вторая версия TCP. В августе 1977 года произошел переломный момент в разработке TCP/IP. Джон Постел (Jon Postel), являющийся одним из разработчиков TCP/IP и Интернет, опубликовал в Internet Engineering Note number 2 ряд комментариев о состоянии TCP. В частности он отметил, что новый протокол пытается выполнять слишком много функций и должен использовать принцип разбиения на уровни. Это замечание Постела привело к созданию архитектуры TCP/IP и разбиению первоначального TCP (Transmission Control Program) на два уровня: Transmission Control Protocol (TCP) на транспортном уровне и Internet Protocol (IP) на сетевом уровне. Процесс разбиения был описан в 1978 году в третьей версии TCP. Первая версия стандартов TCP и IP, используемая в современных сетях, была документирована в 1980 году как TCP version 4 и IP version 4. По этой причине у протокола IP первая версия 4, а не 1. TCP/IP быстро стал набором протоколов для ARPAnet, а позже, в 1983 году — для сети Интернет.

Успех стека протоколов TCP/IP определяется как историческими факторами (протоколы для Интернет), так и техническими характеристиками, включающими интегрированную адресную систему, возможность маршрутизации, независимость от нижележащих технологий LAN, WLAN и WAN, масштабируемость, использование открытых стандартов и универсальность.

### 1.2 Стек протоколов TCP/IP

Стек протоколов TCP/IP был создан раньше модели OSI, поэтому его разработчики не использовали модель OSI для описания архитектуры стека. Они разработали собственную модель, которая имела несколько названий, включая **модель TCP/IP** (Transmission Control Protocol/Internet Protocol), **модель DARPA** (Defense Advanced Research Projects Agency (DARPA или ARPA)) или **модель DOD** (United States Department of Defense).

Так как модель OSI имеет широкое распространение, архитектура TCP/IP часто описывается с использованием названий уровней модели TCP/IP и соответствующих уровней модели OSI.

Модель TCP/IP, так же как и модель OSI, имеет многоуровневую структуру, но для того, чтобы данные от приложения компьютера А были переданы приложению на компьютере B, они должны последовательно пройти 4 уровня: уровень приложений, транспортный уровень, уровень Интернет и уровень доступа к среде.

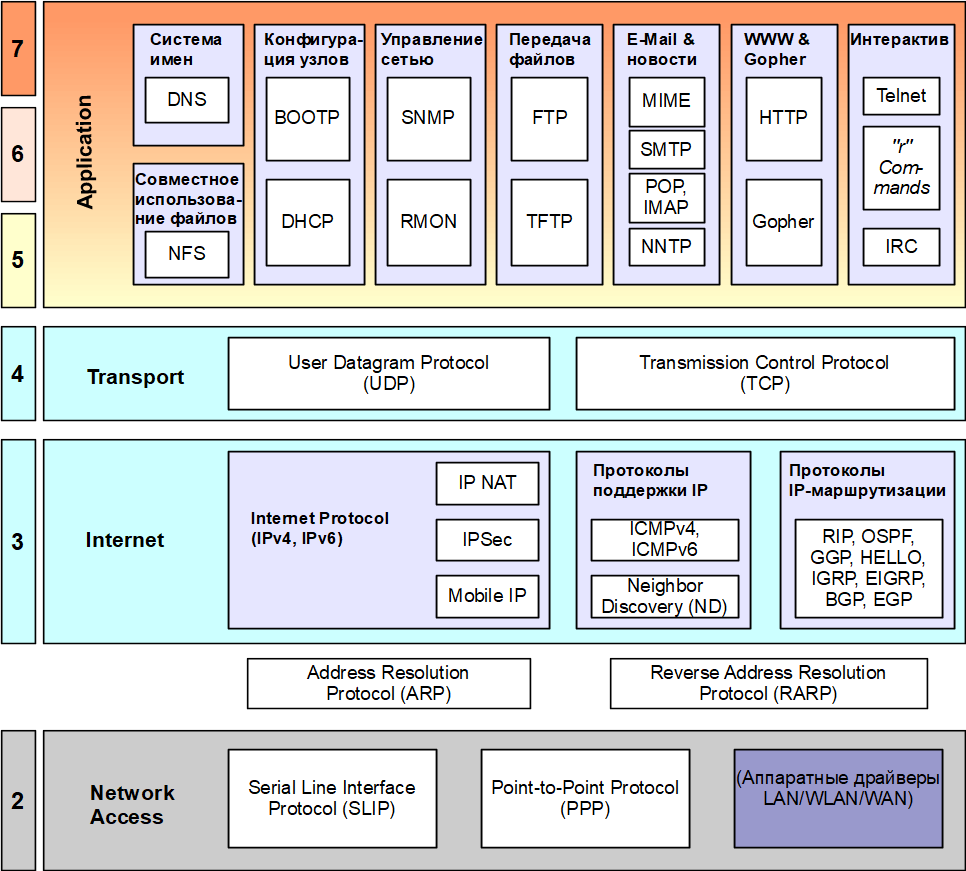
Как показано на рисунке 1.1, трем верхним уровням модели OSI соответствует **уровень приложений** (Application layer) в модели TCP/IP, который включается в себя функции представления, кодирования и контроля над установлением соединения. Существует множество протоколов уровня приложений, из которых самыми распространенными являются FTP, TFTP, HTTP/HTTPs, DHCP, DNS, Telnet, SMTP, POP3, IMAP и др.

**Транспортный уровень** (Transport layer) модели TCP/IP выполняет те же функции, что и одноименный уровень в модели OSI. На этом уровне определены два протокола — TCP и UDP. Протокол TCP (Transmission Control Protocol) обеспечивает надежную доставку сегментов по сети за счет установления логического соединения между отправителем и получателем данных. Протокол UDP (User Datagram Protocol), в отличие от TCP, не устанавливает соединение между отправителем и получателем сообщения и не гарантирует надежную доставку данных.

**Уровень Интернет** (Internet layer) аналогичен по функциям сетевому уровню модели OSI и обеспечивает организацию связи между сетями и подсетями, образующими составную сеть. Основным протоколом уровня Интернет является IP, который выполняет две основные функции — адресацию узлов и выбор наилучшего маршрута до сети назначения (маршрутизацию). Также на этом уровня работают протоколы ICMP, IGMP, протоколы маршрутизации RIP, OSPF, BGP.

**Уровень доступа к среде** (Network access layer) объединяет функции канального и физического уровня модели OSI, обеспечивая физическую передачу данных в сети. Существует множество различных протоколов уровня доступа к сети, из которых самыми распространенными являются Ethernet, IEEE 802.11 (Wi-Fi), PPP, ATM и др.

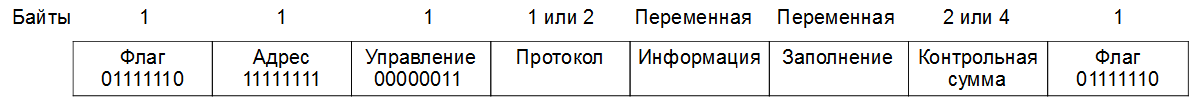




## Протокол PPP

### 2.1 Общий формат кадра PPP

Формат кадра PPP основан на формате кадра HDLC (High-Level Data Link Control, высокоуровневый протокол управления каналом). В отличие от бит-ориентированного протокола HDLC, PPP является байт-ориентированным. Структура общего формата кадра PPP описана в RFC 1662 и показана на рис. 2.2.



**Рис. 2.2** Общий формат кадра PPP

Все кадры PPP начинаются и заканчиваются флагами длиной 1 байт, значение которых равно 01111110 (0×7E). Они используются для синхронизации кадра. Поле **Адрес** (Address) всегда имеет двоичное значение 11111111 (0×FF). Оно означает, что все станции должны принимать этот кадр. Использование такого адреса позволяет избежать назначения станциям индивидуальных адресов, т. к. PPP работает на канале, связывающем только два устройства. За полем адреса следует поле **Управление** (Control), его значение по умолчанию равно 00000011 (0×03). Это число означает ненумерованный кадр.

Четвертое поле кадра PPP — **Протокол** (Protocol). Его размер составляет 1 или 2 байта, а значение определяет тип протокола, данные которого помещены в поле **Информация**. Поле **Информация** (Information) содержит данные протокола, указанного в поле **Протокол**. При необходимости данные могут дополняться специальными символами в поле **Заполнение** (Padding). Поле **Информация** переменной длины. Оно может содержать как 0, так и большее число байтов. Его максимальная длина называется **Maximum Receive Unit** (MRU) и по умолчанию равна 1500 байт. В процессе установки соединения при помощи LCP, стороны могут договориться о значении MRU.

Следом за полем **Информация** располагается поле **Контрольная сумма** (Frame Check Sequence, FCS), длина которого по умолчанию равна 2 байтам. Оно является стандартным 16-битным кодом CRC. В случае необходимости взаимодействующие стороны могут договориться об использовании 4-байтной контрольной суммы, которая представляет собой 32-битный код CRC.

### 2.2 Функционирование канала PPP

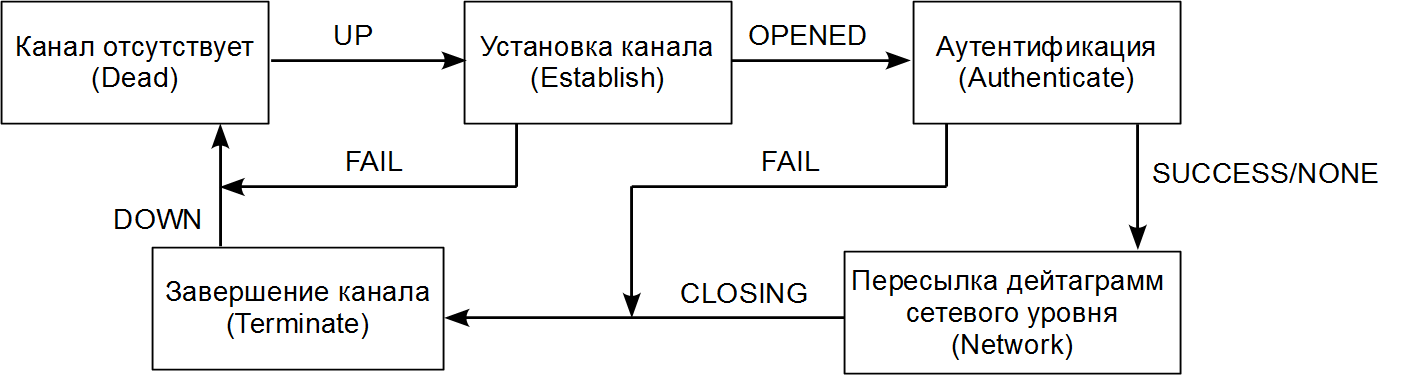
Прежде чем начнется обмен данными по соединению PPP, между двумя устройствами должен быть установлен канал. Для конфигурирования параметров и тестирования канала используется протокол LCP. Конфигурирование параметров выполняется с помощью переговоров. Для этого каждое конечное устройство канала отправляет LCP-пакеты, в которых описывает свои возможности и требования. После установления канала, узлы могут выполнить аутентификацию друг друга, если они об этом договорились ранее.

Для выбора и конфигурирования протоколов сетевого уровня, пакеты которых будут передаваться через канал, используются протоколы NCP. Протоколы NCP сложно описать общими словами, так как каждый из них обладает специфическими свойствами, зависящими от соответствующего протокола сетевого уровня, и поддерживает конфигурационные запросы, характерные только для этого протокола. После установления канала PPP отправляет пакеты NCP для конфигурирования поддерживаемых узлами протоколов сетевого уровня. Как только протокол сетевого уровня сконфигурирован, по каналу могут передаваться его дейтаграммы.

Канал будет оставаться открытым до тех пор, пока явно не будет закрыт LCP- или NCP-пакетами, или пока не произойдет какое-нибудь внешнее событие.

**Диаграмма состояний**

Процесс установления, использования и разъединения канала PPP проходит через несколько состояний, показанных на рис. 2.3.



**Рис. 2.3** Диаграмма состояний соединения РРР

Начальное состояние канала таково: канал отключен (**DEAD**), то есть соединение на физическом уровне отсутствует. После того как физическое соединение установлено, канал переходит в состояние **ESTABLISH** (установка). В этот момент начинаются переговоры о параметрах с помощью протокола LCP. Узлы PPP обмениваются пакетами LCP для выбора параметров соединения. Инициирующий узел предлагает варианты, а отвечающий узел либо соглашается с ним, либо отвергает частично или полностью. Отвечающий узел также может делать свои предложения. Фаза установки канала завершится, как только протокол LCP перейдет в открытое состояние (**OPENED**), т. е. узлы согласуют конфигурационные параметры.

Прежде чем перейти к обмену пакетами протокола NCP, некоторые узлы могут выполнить аутентификацию. По умолчанию аутентификация не обязательна. Однако если в процессе установки канала с помощью LCP стороны договорились о необходимости выполнения аутентификации, то канал переходит в состояние **AUTHENTICATE** (аутентифицировать). В процессе этой фазы выполняется протокол аутентификации. В случае неуспешной аутентификации канал переходит в состояние **TERMINATE** (завершение). После успешной аутентификации канал переходит в состояние **NETWORK** (сеть). В этом состоянии происходит обмен пакетами NCP для настройки сетевого уровня. После того, как NCP перейдет в открытое состояние, РРР начинает передачу пакетов соответствующего протокола сетевого уровня. В данном состоянии трафик на канальном уровне состоит из комбинации пакетов LCP, NCP и протокола сетевого уровня.

Когда передача данных закончена, канал переходит к фазе **TERMINATE** (завершение), а затем снова в состояние **DEAD** (отключен), когда физическое соединение разрывается.

Канал PPP может быть закрыт в любое время по требованию пользователя или по другим причинам: из-за потери несущей, неудачной аутентификации, недостаточного качества канала, по истечении допустимого периода времени неиспользования канала.

Для закрытия канала используется протокол LCP. Он выполняет это путем обмена сообщениями **Terminate-Request** и **Terminate-Ack** между взаимодействующими узлами. Когда канал закрывается, PPP информирует об этом протоколы сетевого уровня, таким образом они могут выполнить соответствующие действия.

Закрытие канала с помощью LCP является достаточным. Отправлять поток пакетов **Terminate** каждому протоколу NCP нет необходимости. Более того, тот факт, что один из NCP закрылся, не является достаточной причиной для закрытия канала PPP, даже если этот NCP в тот момент был в открытом состоянии единственным.

### 2.3 Link Control Protocol (LCP)

Протокол LCP выполняет три важных функции:

* Конфигурирование канала: процесс установления и настройки параметров канала.
* Обслуживание канала: процесс управления и диагностики открытого канала.
* Завершение канала: процесс закрытия существующего канала, если он больше не требуется или когда разрывается физическое соединение.

Каждая из этих функций соответствует одному из состояний соединения PPP. Конфигурирование канала выполняется в состоянии **ESTABLISH**. Обслуживание канала выполняется, когда LCP переходит в открытое состояние. Завершение канала происходит в состоянии **TERMINATE**.

Протокол LCP выполняет свои функции путем обмена сообщениями между взаимодействующими устройствами. Эти сообщения называются пакетами LCP. Пакет LCP инкапсулируется в поле **Информация** кадра PPP, в поле **Протокол** которого установлено значение 0хС021.

Общий формат пакет LCP показан на рисунке рис. 2.4.



**Рис. 2.4** Общий формат пакета LCP

Поле **Код** (Code), длиной 1 байт, описывает тип пакетов LCP. Однобайтное поле **Идентификатор** (Identifier) позволяет определять пары запросов и ответов. Поле **Длина** (Length) имеет длину 2 байта и содержит длину пакета LCP, с учетом полей **Код**, **Идентификатор**, **Длина** и **Данные**. Поле **Данные** (Data) переменной длины. Его формат определяется типом пакета LCP, указанным в поле **Код**.

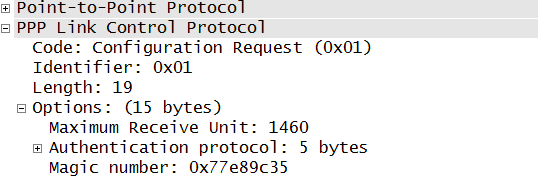
**Установление канала**

Установление и конфигурирование канала выполняются автоматически путем обмена взаимодействующими узлами конфигурационными пакетами. Узел-инициатор отправляет пакет **Configure-Request**. Он может содержать переменное количество параметров настройки (опций), о которых инициатор желает договориться.

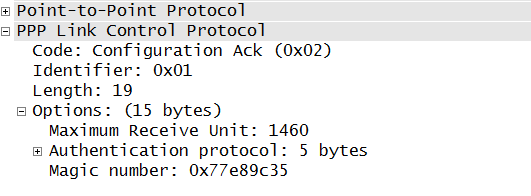
Протоколом LCP согласуются только те опции, которые не зависят от конкретных протоколов сетевого уровня. В RFC 1661 определены 6 конфигурационных опций, которые инициатор может указать в запросе **Configure-Request**:

* **Maximum Receive Unit (MRU)**: позволяет указывать максимальный размер передаваемой по каналу дейтаграммы.
* **Authentication Protocol**: позволяет указывать тип протокола аутентификации, если обе стороны желают его использовать. По умолчанию аутентификация не требуется.
* **Quality Protocol**: позволяет указывать тип протокола оценки качества канала. По умолчанию оценка качества канала отключена.
* **Magic Number**: позволяет определять закольцованные каналы или другие нарушения работы канального уровня.
* **Protocol Field Compression**: позволяет договориться о сжатии поля **Протокол** кадра PPP.
* **Address and Control Field Compression (ACFC)**: позволяет договориться о сжатии полей **Адрес** и **Управление** кадра PPP.

Если отвечающий узел соглашается со всеми указанными инициатором параметрами, он отправляет пакет **Configure-Ack**. Как только пакет **Configure-Ack** получен инициатором, протокол LCP переходит в открытое состояние и фаза установления канала завершается.



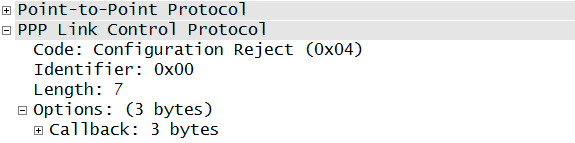
**Рис. 2.5** Пакет Configure-Request



**Рис. 2.6** Пакет Configure-Ack

Если отвечающий узел согласен не со всеми предлагаемыми параметрами, он отправляет узлу-инициатору сообщение **Configure-Nak**, в котором содержится список неподтвержденных параметров. Узел-инициатор обрабатывает ответ и отправляет пакет **Configure-Request**, содержащий новые конфигурационные параметры.

Если отвечающий узел полностью отвергает предложенные параметры, он отправляет сообщение **Configure-Reject**. Также как и в случае получения пакета **Configure-Nak**, узел-инициатор повторяет попытку договориться о параметрах канала и отправляет пакет **Configure-Request**, содержащий новые конфигурационные параметры.



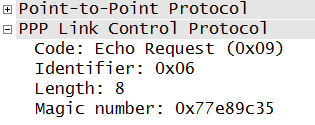
**Рис. 2.7** Пакет Configure-Reject

В фазе установления канала, взаимодействующие узлы будут игнорировать все не-LCP пакеты. Узлы вернутся в фазу установления канала, если в фазе аутентификации или пересылки дейтаграмм сетевого уровня одним из них будет получен пакет **Configure-Request.**

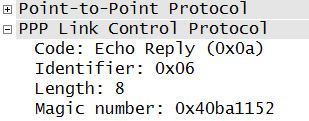
**Обслуживание канала**

Как только переговоры о параметрах канала успешно завершились, LCP переходит в открытое состояние и передает управление соответствующим протоколам аутентификации и/или NCP. В этой фазе пакеты LCP могут использоваться для управления, поиска неисправностей в канале или тестирования производительности. Пакеты **Code-Reject** и **Protocol-Reject** используются для обратной связи, когда одна из сторон канала получила пакет LCP с неизвестным кодом или кадр PPP с неподдерживаемым типом протокола.

Для исследования канала в обоих направлениях используются пакеты **Echo-Request** и **Echo-Reply**. Пакет **Discard-Request** используется для исследования канала в направлении от отправителя к получателю.



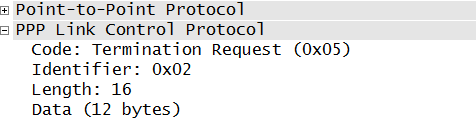
**Рис. 2.8** Пакет Echo-Request



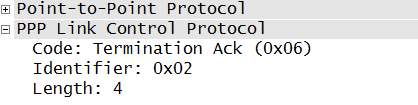
**Рис. 2.9** Пакет Echo-Reply

**Завершение канала**

Для завершения работы канала одна из его сторон отправляет пакет **Terminate-Request**. Другая сторона отвечает пакетом **Terminate-Ack**.



**Рис. 2.10** Пакет Terminate-Request



**Рис. 2.11** Пакет Terminate-Ack

### 2.4 Network Control Protocol (NCP)

После завершения фазы установления канала протоколом LCP и аутентификации (которая необязательна и может не выполняться), независимо друг от друга должны быть сконфигурированы протоколы сетевого уровня, такие как IP, IPX и т. п. Конфигурирование параметров протоколов сетевого уровня выполняется протоколами управления сетью (NCP). Для каждого протокола сетевого уровня, пакеты которого передаются по каналу PPP, используется собственный протокол NCP. Каждый из протоколов NCP обладает специфическими свойствами, зависящими от соответствующего протокола сетевого уровня, и поддерживает конфигурационные запросы, характерные только для этого протокола. Каждый из протоколов NCP может переходить в открытое состояние и закрываться в любое время.

**Таблица 2.1**. Стандарты протокола NCP

|  |  |  |
| --- | --- | --- |
| **Номер RFC** | **Название RFC** | **Описание** |
| 1332 | The PPP Internet Protocol Control Protocol (IPCP) | NCP для протокола Internet Protocol (IP) |
| 1377 | The PPP OSI Network Layer Control Protocol (OSINLCP) | NCP для набора протоколов сетевого уровня OSI таких как CNLP, ES-IS и IS-IS |
| 1378 | The PPP AppleTalk Control Protocol (ATCP) | NCP для протокола AppleTalk |
| 1552 | The PPP Internetworking Packet Exchange Control Protocol (IPXCP) | NCP для протокола Novell Internetworking Packet Exchange (IPX) |
| 2043 | The PPP SNA Control Protocol (SNACP) | NCP для протокола IBM Systems Network Architecture (SNA) |
| 2097 | The PPP NetBIOS Frames Control Protocol (NBFCP) | NCP для кадров NetBIOS |
| 2472 | IP Version 6 over PPP (IPCPv6) | NCP для протокола IPv6 |

Аналогично LCP, каждый протокол NCP выполняет установление, обслуживание и завершение канала. Для выполнения этих функций обе стороны канала обмениваются пакетами протокола NCP. Процесс установки канала и переговоров о параметрах соответствующего протокола сетевого уровня выполняется с помощью пакетов **Configure-Request**, **Configure-Ack**, **Configure-Nak** и **Configure-Reject**, аналогичных LCP. Конфигурационные параметры, передаваемые в этих пакетах, зависят от используемого протокола NCP. Протокол NCP переходит в открытое состояние после согласования параметров, т. е. после получения инициатором пакета **Configure-Ack**. Пакет **Code-Reject** отправляется в том случае, когда одна из сторон канала получила пакет NCP с неизвестным кодом. Для завершения канала одна из его сторон отправляет пакет **Terminate-Request**, на который другая сторона отвечает пакетом **Terminate-Ack**.

**Протокол конфигурирования IPv4 (IPCP)**

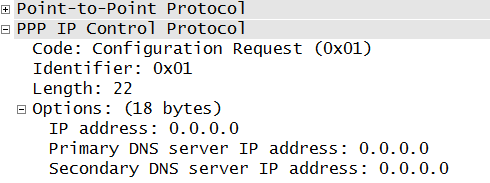
Рассмотрим в качестве примера протокол NCP для IPv4 (IPCP). Протокол управления IP (Internet Protocol Control Protocol, IPCP) предназначен для конфигурирования протокола IPv4 на обоих концах канала PPP. Обмен IPCP-пакетами аналогичен LCP и начинается в состоянии обмена дейтаграммами сетевого уровня (**NETWORK**). IPCP отличается от LCP следующим:

* Поле **Протокол** кадра PPP. Пакет IPCP инкапсулируется в поле **Информация** кадра PPP, поле **Протокол** которого имеет значение 0х8021.
* Поле **Код** пакета IPCP. Используются только коды с номерами 0-7: **Configure-Request**, **Configure-Ack**, **Configure-Nak**, **Configure-Reject**, **Terminate-Request**, **Terminate-Ack**, **Code-Reject**.

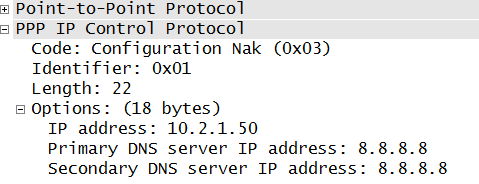
Процесс установки канала и переговоров о параметрах протокола IP выполняется с помощью пакетов **Configure-Request**, **Configure-Ack**, **Configure-Nak** и **Configure-Reject**.

Для протокола IP определены две конфигурационные опции, которые могут быть указаны в пакете IPCP **Configure-Request**:

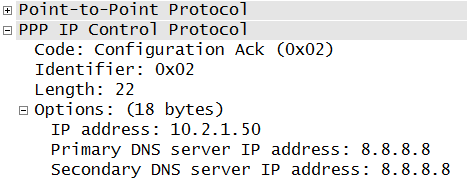
1. **IP-Compression-Protocol** (протокол сжатия IP). Данный параметр позволяет договориться об использовании конкретного протокола сжатия заголовков протоколов TCP и IP. По умолчанию сжатие не требуется.
2. **IP-Address** (IP-адрес). Данный параметр служит для ведения переговоров об IP-адресе, который будет назначен локальной стороне канала. Он позволяет отправителю запроса **Configure-Request** указать желаемый IP-адрес или запросить противоположную сторону выдать ему IP-адрес. Противоположная сторона может предоставить данную информацию, отвергнув этот параметр с помощью **Configure-Nak** и указав правильный IP-адрес. Также параметр позволяет договориться об IP-адресах первичного и вторичного DNS-серверов, которые будут использоваться на локальной стороне. Когда отвечающий узел соглашается с параметрами, он отправляет пакет **Configure-Ack**.



**Рис. 2.12** Запрос IP-адресов протоколом IPCP (сообщение Configure-Request)



**Рис. 2.13** Назначение IP-адресов локальной стороне канала (сообщение Configure-Nak)



**Рис. 2.14** Подтверждение назначенных IP-адресов (сообщение Configure-Ack)

После того, как протокол IPCP перейдет в открытое состояние, по каналу PPP смогут передаваться любые IP-пакеты. Максимальная длина IP-пакета соответствует максимальной длине поля **Информация** кадра РРР. IP-дейтаграммы большего размера будут фрагментированы. Если необходимо избежать фрагментирования и последующего сбора пакетов, то следует использовать механизм **Path MTU discovery**. Фрагментация пакетов IPv4 и работа процесса Path MTU discovery будет рассмотрена далее в этом курсе.

### 2.5 Протоколы аутентификации PPP

Аутентификация в протоколе PPP не является обязательной и выполняется только по согласованию сторон. Взаимодействующие устройства договариваются об используемом протоколе аутентификации в процессе переговоров LCP. В случае успешной договоренности, аутентификация выполняется сразу же после установления канала. Продолжение настройки канала при этом будет возможно только после успешной аутентификации.

|  |
| --- |
| **Аутентификация** (Authentication) — сервис безопасности, который обеспечивает подтверждение того, что информация получена от законного источника и получатель является требуемым.  Идентификация (Identification) — сервис, с помощью которого определяются уникальные свойства пользователей, которые позволяют отличать их друг от друга, и способы, с помощью которых пользователи указывают свои идентификации информационной системе. Идентификация тесно связана с аутентификацией. |

Первоначально в стеке протоколов PPP были определены два протокола аутентификации: Password Authentication Protocol (PAP) и Challenge Handshake Authentication Protocol (CHAP). Они описаны в RFC 1334 «PPP Authentication Protocols».

В настоящее время PPP поддерживает несколько протоколов аутентификации:

* Password Authentication Protocol (PAP);
* Challenge Handshake Authentication Protocol (CHAP);
* Microsoft Challenge-Handshake Authentication Protocol version 1.0 (MS-CHAP v1);
* Microsoft Challenge Handshake Authentication Protocol version 2.0 (MS-CHAP v2);
* Extensible Authentication Protocol — Transport Level Security (EAP-TLS);
* Protected Extensible Authentication Protocol (PEAP).

В PPP используется следующая терминология для описания аутентифицируемых узлов:

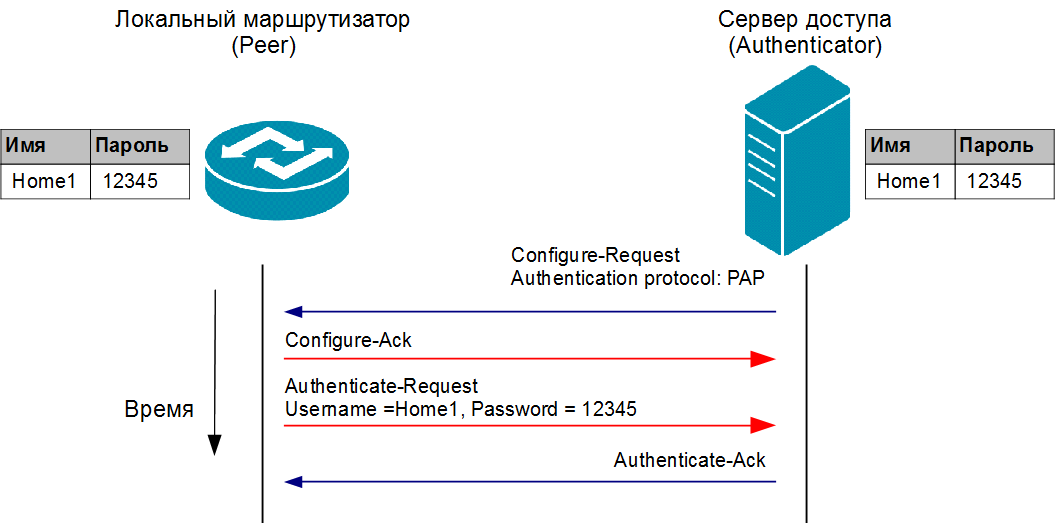
* **Аутентификатор** (authenticator) — конец канала, требующий выполнения аутентификации. Аутентификатор определяет протокол аутентификации, который будет указан в пакетах **Configure-Request**, передаваемых в фазе установления канала.
* **Одноранговый узел** (peer) — другая сторона канала «точка-точка», чья подлинность проверяется аутентификатором.

### 2.5.1 Протокол Password Authentication Protocol (PAP)

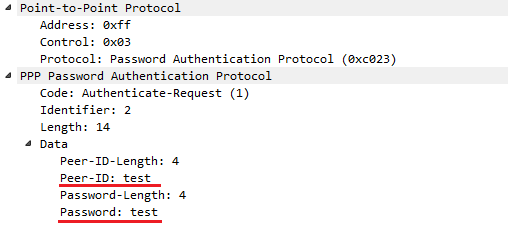
Протокол PAP — простой протокол двухстороннего рукопожатия (2-way handshake). Процесс аутентификации состоит из двух шагов:

1. **Запрос аутентификации**. Узел отправляет аутентификатору запрос **Authenticate-Request**, который содержит имя пользователя и пароль.
2. **Ответ на запрос аутентификации**. Аутентификатор проверяет полученные имя пользователя и пароль в локальной базе данных или пересылает их серверу аутентификации (например, RADIUS или TACACS+), и если они достоверны, отправляет узлу сообщение **Authenticate-Ack**. Аутентификация успешно завершается и начинается фаза конфигурирования протоколов сетевого уровня. Если аутентификация не успешна, аутентификатор отправляет узлу сообщение **Authenticate-Nak** и соединение завершается.

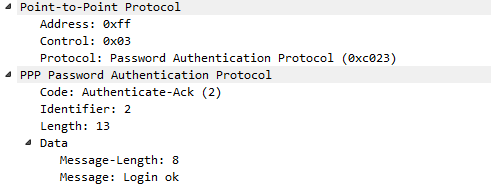
В процесс аутентификации пароль передается в открытом виде, т. е. он не шифруется. В связи с этим, аутентификация PAP не является безопасной, т. к. не обеспечивает защиты от различного рода атак.



**Рис. 2.15** Аутентификация PAP



**Рис. 2.16** Запрос аутентификации PAP (Authenticate-Request)



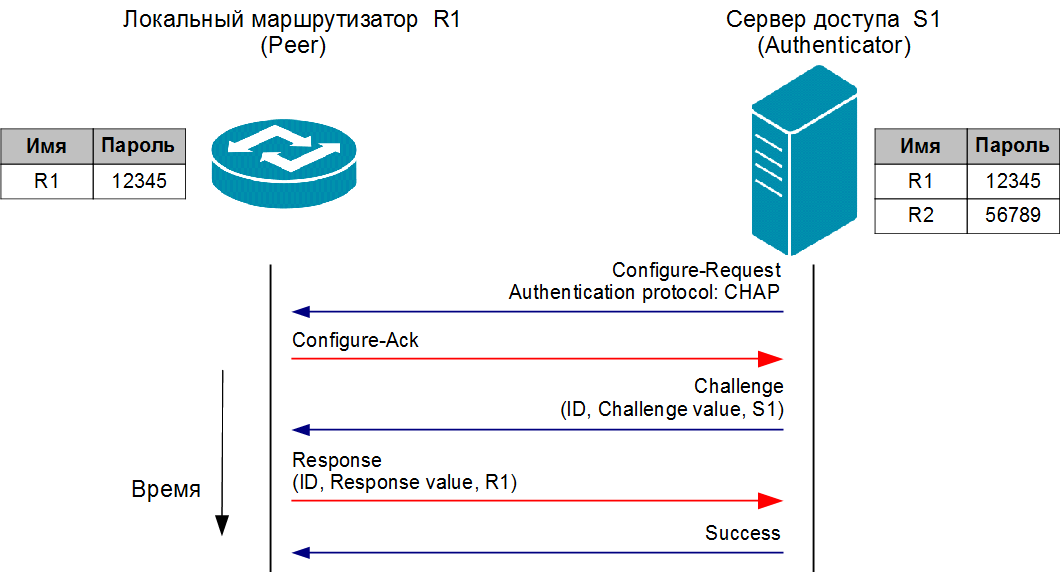
**Рис. 2.17** Ответ на запрос аутентификации PAP (Authenticate-Ack)

### 2.5.2 Протокол Challenge Handshake Authentication Protocol (CHAP)

Протокол СНАР используется для периодической проверки аутентификации противоположной стороны с помощью трехстороннего рукопожатия (3-way handshake). Протокол выполняется после установления канала и может быть повторно использован в любое время после того, как канал установлен. Первоначально CHAP был описан в RFC 1334, а позднее дополнен в RFC 1994.

|  |
| --- |
| **Алгоритм DES** (Data Encryption Standard) — алгоритм симметричного шифрования, длина блока равна 64 битам, длина ключа равна 56 битам.  Атаки повторного использования (replay-атаки) — пассивный захват данных с последующей их пересылкой целевой системе для получения несанкционированного доступа.  Хэш-функция — функция, отображающая данные произвольной длины в строку битов фиксированной длины, которая может использоваться для аутентификации исходных данных.  Хэш-код — результат, создаваемый хэш-функцией. |

Процесс аутентификации СНАР показан на рисунке 2.18.



**Рис. 2.18** Аутентификация CHAP

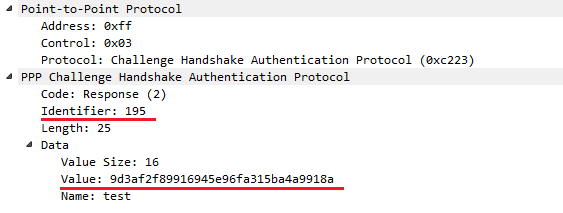
Аутентификация состоит из следующих шагов:

* После завершения фазы установления канала, аутентификатор посылает узлу сообщение **Challenge**. Оно содержит идентификатор (Identifier) этого сообщения и вызов (Challenge Value) — произвольную строку символов переменной длины.



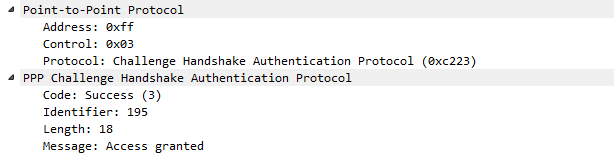
**Рис. 2.19** Сообщение Challenge CHAP

* Узел-получатель выполняет одностороннюю хэш-функцию (обычно Message Digest 5 (MD5)) над информацией, полученной из сообщения **Challenge** (идентификатором и вызовом) и паролем. Вычисленное значение он отправляет аутентификатору в сообщении **Response**. Идентификатор сообщения **Response** аналогичен идентификатору сообщения **Challenge**.



**Рис. 2.20** Сообщение Response CHAP

* Аутентификатор сравнивает полученное значение с собственным вычисленным значением хэш-функции. Хэш-функция на стороне аутентификатора выполняется над идентификатором и вызовом оригинального сообщения **Challenge** и паролем. Поэтому у аутентификатора и узла должны быть одинаковые пароли. Если вычисленное и полученное значения хэш-функции совпали, аутентификация считается успешной и аутентификатор отправляет сообщение **Success**. В противном случае соединение сбрасывается и отправляется сообщение **Failure**.



* **Рис. 2.21** Сообщение Success CHAP
* Через произвольные интервалы времени аутентификатор отправляет новый **Challenge** и шаги 1–3 повторяются.

СНАР обеспечивает защиту от replay-атак благодаря использованию при каждой отправке сообщения **Challenge** уникального значения идентификатора и случайного набора символов переменной длины. Таким образом, нет возможности воспользоваться перехваченной информацией для получения неавторизованного доступа к сети. Повторная отправка сообщений **Challenge** в течение сессии CHAP предназначена для ограничения времени на проведения любых единичных атак. Аутентификатор управляет частотой и временем отправки сообщений **Challenge**.

Метод аутентификации CHAP основан на том, что оба участника знают некий секрет (пароль), который никогда не отправляется по каналу. Хотя аутентификация односторонняя, стороны могут договориться о применении CHAP в обоих направлениях и использовать одинаковый пароль для обеспечения взаимной аутентификации. Т. к. CHAP может использоваться для аутентификации множества различных устройств, поле **Имя** (Name) в сообщениях **Challenge** и **Response** может служить индексом для поиска правильного пароля в больших таблицах, хранящих их. Также это позволяет создавать для каждого устройства более одной пары имя/пароль и изменять используемый пароль в любое время в течение сессии.

Аутентификация CHAP имеет свои недостатки, в частности требует, чтобы пароль хранился в открытом виде. При этом она устраняет одну из уязвимостей PAP — пароль не передается по сети в открытом виде, а используется вызов и хэширование ответа, что позволяет избежать replay-атак. Также, в отличие от PAP, аутентификация CHAP может выполняться несколько раз за сессию. В 1996 году IETF обновила стандарт, описывающий аутентификацию PPP, и оставила в нем только CHAP. Новый стандарт получил название RFC 1994 «PPP Challenge Handshake Authentication Protocol (CHAP)». RFC 1334 в настоящее время является устаревшим.