Глава 2 . Стек протоколов TCP/IP

2.1. Предисловие к главе

Семейство протоколов TCP/IP, часто именуемое ***стеком TCP/IP***, стало промышленным стандартом де-факто для обмена данными между процессами распределенного приложения и поддерживается всеми без исключения операционными системами общего назначения. Обширная коллекция сетевых протоколов и служб, называемая стеком TCP/IP намного больше, чем сочетание двух основных протоколов давшее ей имя. Тем не менее, эти протоколы являются основой стека TCP/IP: ***TCP (Transmission Control Protocol)*** обеспечивает надежную доставку данных в сети; ***IP (Internet Protocol)*** организует маршрутизацию сетевых передач от отправителя к получателю и отвечает за адресацию сетей и компьютеров.

В настоящее время существует шесть групп стандартизации координирующих TCP/IP: ISOC (Internet Society, http://www.isoc.org), IAB (Internet Architecture Board, http://www.iab.org), IETF (Internet Engineering Task Force, http://www.ietf.org), IRTF (Internet Research Task Force, http://www.irtf.org), ISTF (Internet Societal Task Force, http://www.istf.org), ICANN (Internet Corporation for Assigned Names and Numbers, http://www.icann.org). Наиболее важной организацией из перечисленных является IETF – проблемная группа по проектированию Internet, поскольку именно она занимается поддержкой документов именуемых ***RFC (Request for Comments)***, в которых описаны все правила и форматы всех протоколов и служб TCP/IP в сети Internet.

Всем RFC присвоены номера. Например, специальный документ “Официальные стандарты протоколов сети Internet” имеет номер RFC 2700. Другой важный документ RFC 2026 определяет порядок создания самого документа RFC и процедуры, которые должны быть им пройдены для превращения в официальный стандарт группы IETF. Если есть документы, имеющее одно название, но разные номера, то документ с самым большим номером считается текущей версией.

Более подробно с историей создания TCP/IP, с назначением групп стандартизации и процедурами создания и утверждения документов RFC можно ознакомиться в [5,6].

В этой главе рассматриваются основные компоненты стека протоколов TCP/IP, необходимые для построения распределенного приложения.

2.2. Структура TCP/IP

Так как архитектура TCP/IP была разработана задолго до модели ISO/OSI, то неудивительно, что конструкция TCP/IP несколько отличается от эталонной модели. На рисунке 2.2.1. изображены уровни обеих моделей и устанавливается их соответствие. Уровни моделей похожи, но не идентичны. Структура TCP/IP является более простой: в ней не выделяются Физический, Канальный, Сетевой и Представительский уровни. В общем и целом Транспортные уровни обеих моделей соответствуют друг другу, но есть и

некоторые различия. Например, некоторые функции Сеансового уровня модели ISO/OSI берет на себя Транспортный уровень TCP/IP. Содержимое Сетевого уровня модели ISO/OSI тоже примерно соответствует Межсетевому уровню TCP/IP. В большей или меньшей степени Прикладной уровень TCP/IP соответствует трем уровням Сетевому, Представительскому и Прикладному модели ISO/OSI, а Уровень доступа к сети – совокупности Физического и Канального уровней.

Рисунок 2.2.1. Уровни моделей ISO/OSI и TCP/IP

В дальнейшем при описании протоколов стека TCP/IP будет использоваться и названия уровней модели ISO/OSI, если это помогает определить более точное место протокола в иерархии.

2.3. Протоколы Уровня доступа к сети

На Уровне доступа к сети задействованы протоколы для создания локальных сетей (***LAN***, Local-Area Networks) и для соединения с глобальными сетями (***WAN***, Wide-Area Networks). Работа протоколов этого уровня регулируются семейством стандартов ***IEEE 802*** (Institute of Electrical

and Electronic Engineers), включающее помимо прочих компоненты: IEEE 802.1 по межсетевому обмену; IEEE 802.2 по управлению логическим соединением (***LLC***, Logical Link Control); IEEE 802.3 по управлению доступом к среде (***MAC***, Media Access Control); IEEE 802.4 по множественному доступу с контролем несущей и обнаружением конфликтов (***CSMA/CD***, Carrier Sense Multiple Access with Collision Detection). Одной из основных характеристик протоколов канального уровня является ***максимальная единица передачи данных MTU*** (Maximum Transmission Unit), которая определяет максимальную длину в байтах данных передаваемых в одном кадре. От значения MTU зависит скорость передачи по каналу. Если IP-модулю требуется отправить данные (дейтаграмму) имеющие дину большую MTU, то он производит фрагментацию разбивая дейтаграмму на части имеющие длину меньшую, чем MTU. В таблице 2.3.1 приведены типичные значения MTU для некоторых сетей.

Таблица 2.3.1

Сеть	**MTU (байты)**
FDDI	4464
Ethernet	1500
IEEE802.3/802.2	1492
X.25	576
SLIP, PPP (с минимальной задержкой)	256

Протокол Ethernet. Термин Ethernet обычно связывают со стандартом опубликованным в 1982 г. совместно корпорациями DEC, Intel и Xerox (DIX). На сегодняшний день это наиболее распространенная технология локальных сетей. Ethernet применяет метод доступа CSMA/CD, использует 48-битную адресацию (стандарт IEE EUI-64) и обеспечивает передачу данных до 1 гигабита в секунду. Максимальная длина кадра, передаваемая в сети Ethernet составляет 1518 байт, при этом сами данные могут занимать от 46 до 1500 байт. Физически функции протокола Ethernet реализуются сетевой картой (***NIC***, Network Interface Card), которая может быть подключена к кабельной системе с фиксированным MAC-адресом производителя (в соответствии со стандартом ICANN).

Протокол SLIP(Serial Line IP). Аббревиатурой SLIP обозначают межсетевой протокол для последовательного канала. Раньше SLIP использовался для подключения домашних компьютеров к Internet через последовательный порт RS-232. Протокол использует простейшую инкапсуляцию кадра и имеет ряд недостатков: хост с одной стороны должен знать IP-адрес другого, т.к. SLIP не дает возможности сообщить свой IP-адрес; если линия задействована SLIP, то она не может быть использована никаким другим протоколом; SLIP не добавляет контрольной информации к пакету передаваемой информации – весь контроль возложен на протоколы

более высокого уровня. Ряд недостатков были исправлены в новой версии протокола именуемой CSLIP (Compressed SLIP).

Протокол PPP (Point-to-Point Protocol). PPP – универсальный протокол двухточечного соединения: поддерживается TCP/IP, NetBEUI, IPX/SPX, DECNet и многими стеками протоколов. Протокол может применяться для технологии ISDN (Integrated Services Digital Network) и SONET (Synchronous Optical Network). PPP поддерживает многоканальные реализации: можно сгруппировать несколько каналов с одинаковой пропускной способностью между отправителем и получателем. Кроме того, PPP обеспечивает циклический контроль для каждого кадра, динамическое определение адресов, управление каналом. В настоящее время это наиболее широко используемый протокол для последовательного канала, обеспечивающий соединение компьютера с сетью Internet и практически вытеснил протокол SLIP.

2.4. Протоколы Межсетевого уровня

Важнейшими протоколами Межсетевого уровня TCP/IP являются: ***IP*** (Internet Protocol), ***ICMP*** (Internet Control Message Protocol), ***ARP*** (Address Resolution Protocol), ***RARP*** (Reverse ARP).

Протокол IP. В семействе протоколов TCP/IP протоколу IP отведена центральная роль. Его основной задачей является доставка ***дейтограмм*** (так называется единица передачи данных в терминологии IP). При этом протокол по определению является ***ненадежным*** и ***не поддерживающим соединения***.

Ненадежность протокола IP обусловлена тем, что нет гарантии, что посланная узлом сети дейтаграмма дойдет до места назначения. Сбой, произошедший на любом промежуточном узле сети, может привести к уничтожению дейтаграмм. Предполагается, что необходимая степень надежности должна обеспечиваться протоколами верхних уровней.

IP не ведет никакого учета очередности доставки дейтаграмм: каждая дейтаграмма обрабатывается независимо от остальных. Поэтому очередность доставки может нарушаться. Предполагается, что учетом очередности дейтаграмм должен заниматься протокол верхнего уровня.

Если протоколы Уровня доступа к сети при передаче данных используют MAC-адреса, то на Межсетевом уровне применяется IP-адресация. Главной особенностью IP-адреса является его независимость от физической устройства, подключенного к сети. Это дает возможность на уровне IP одинаковым образом обрабатывать данные, полученные или отправленные с помощью модема, сетевой карты или любого другого устройства, поддерживающего интерфейс протокола IP. Все устройства, имеющие IP-адрес, в терминологии протокола IP называются ***хостами*** (host).

IP-адрес представляет собой последовательность из 32 битов. Причем старшие (левые) биты этой последовательности отводятся для адреса сети, а младшие (правые) – для адреса хоста в этой сети. Для записи IP-адреса, как

правило, используются четыре десятичных числа, разделенных точкой. Каждое десятичное число является десятичным представлением 8 битов (***октет*** в терминологии TCP/IP) адреса. На рисунке 2.4.1 разобран пример перевода IP-адреса из двоичного формата в десятичный.

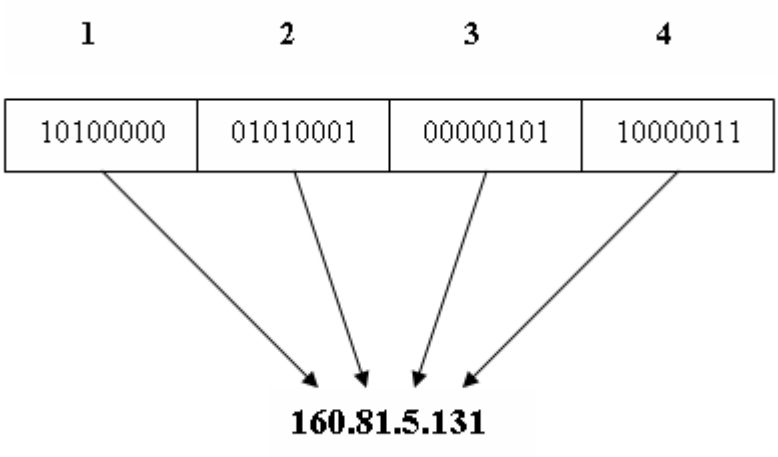

Рисунок 2.4.1. Представление IP-адреса в десятичном формате

Количество бит отведенных для адреса сети и адреса хоста определяется ***моделью адресации***. Существует две модели адресации: ***классовая*** и ***бесклассовая***. В классовой модели адресации все адреса подразделяются на пять классов: A, B, C, D, E. Принадлежность к классу определяется старшими битами адреса. На рисунке 2.4.2 приведены форматы адресов для всех классов.

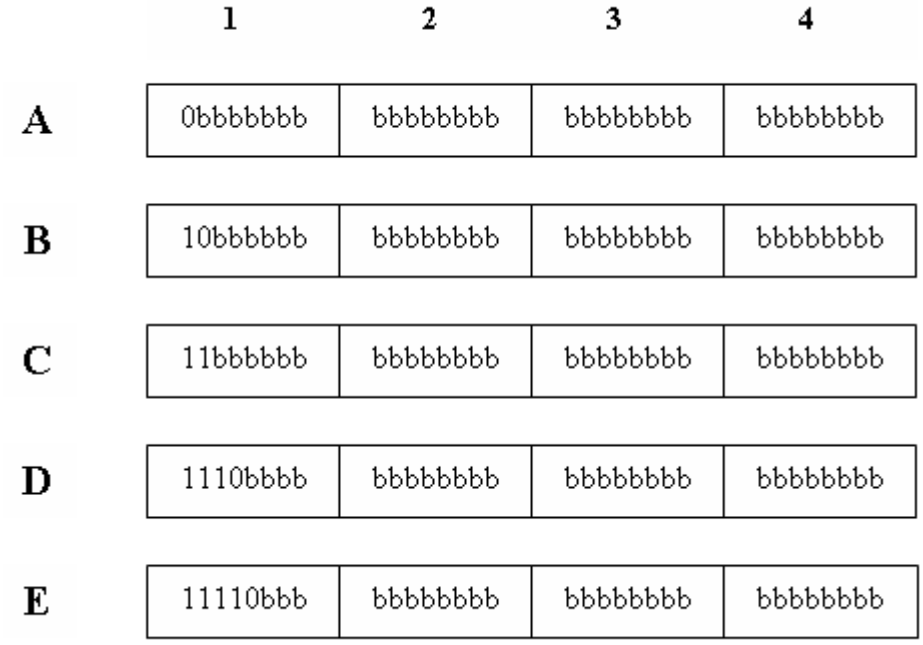

Рисунок 2.4.2 Форматы IP-адресов в классовой модели адресации

Классы D и E имеют специальное назначение: D – предназначен для использования групповых адресов, позволяющих отправлять сообщения группе хостов; E – исключительно для экспериментального применения. Более подробно о применении адресов классов D и E можно ознакомиться в [5,6]. Распределение октетов IP-адреса на адрес сети и адрес хоста для классов A, B и C приводится на рисунке 2.4.3. Закрашенные октеты обозначают часть IP-адреса отведенную для адреса сети, не закрашенные – часть адреса, используемую для адресов хостов. Правый столбец показывает формат адресов в десятичном виде: символом n обозначается сетевая часть адреса, символом h – часть адреса для идентификации хоста.

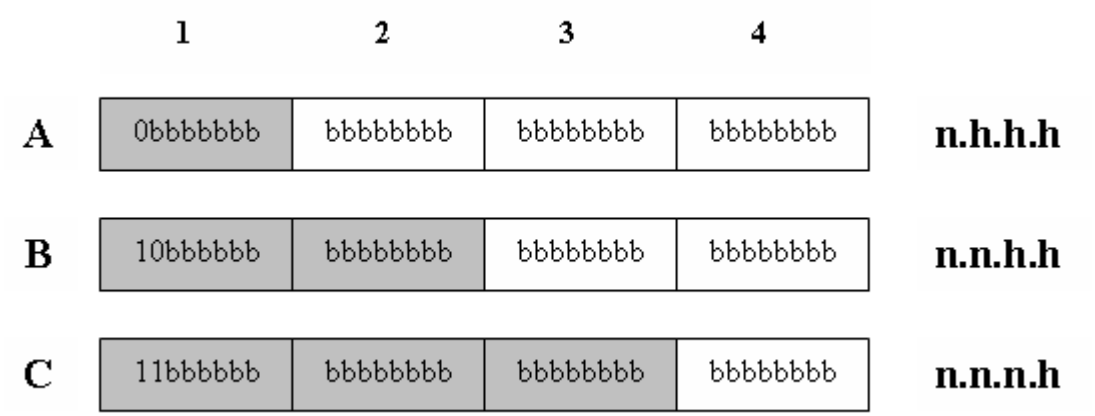

Рисунок 2.4.3. Распределение разрядов IP-адреса на адрес сети и адрес хоста для классовой модели IP-адресов

Таблица 2.4.1

Класс	**Диапазон адресов**	**Диапазон частных адресов**
A	0.0.0.0 – 127.255.255.255	10.0.0.0 – 10.255.255.255
B	128.0.0.0 – 191.255.255.255	172.16.0.0 – 172.31.255.255
C	192.0.0.0 – 223.255.255.255	192.168.0.0 – 192.168.255.255
D	224.0.0.0 – 239.255.255.255	не предусмотрен
E	240.0.0.0 – 247.255.255.255	не предусмотрен

Для каждого класса адресов зарезервирован диапазон, используемый для частного применения (это оговаривается в документе RFC 1918). Эти адреса предназначены для неконтролируемого использования в организациях. Уникальность этих адресов в сети Internet не гарантируется и поэтому их маршрутизация в сети Internet не возможна. Кроме того, адреса вида 127.n.n.n предназначены для выполнения возвратного тестирования (***loopback testing***). Существует некоторая путаница с применением адресов состоящих из всех нулей или единиц. До выхода документа RFC 1878 (1995 г.) не разрешалось использование в качестве адреса хоста нулевую последовательность битов (этот адрес считался адресом самой сети) и

последовательность битов, состоящую из одних единиц (этот адрес использовался для широковещательных сообщений в сети). RFC 1878 разрешает использование таких адресов. Диапазоны IP- адресов для каждого класса сведены в таблице 2.4.1.

Применение классовой модели адресации не всегда удобно. Альтернативой классовой модели является ***бесклассовая междоменная маршрутизация*** – ***CIDR*** (Classless Inter-Domain Routing). CIDR позволяет произвольным образом назначать границу сетевой и хостовой части IP-адреса. Для этого каждому IP-адресу прилагается 32-битовая маска, которую часто называют ***маской сети*** (net mask) или ***маской подсети*** (subnet mask). Сетевая маска конструируется по следующему правилу: на позициях, соответствующих адресу сети, биты установлены; на позициях, соответствующих адресу хоста, биты сброшены. Установка маски подсети осуществляется при настройке протоколов TCP/IP на компьютере. Вычисление адреса сети выполняется с помощью операции конъюнкции между IP-адресом и маской подсети. На рисунке 2.4.4. разобран пример вычисления адреса сети с помощью маски подсети.

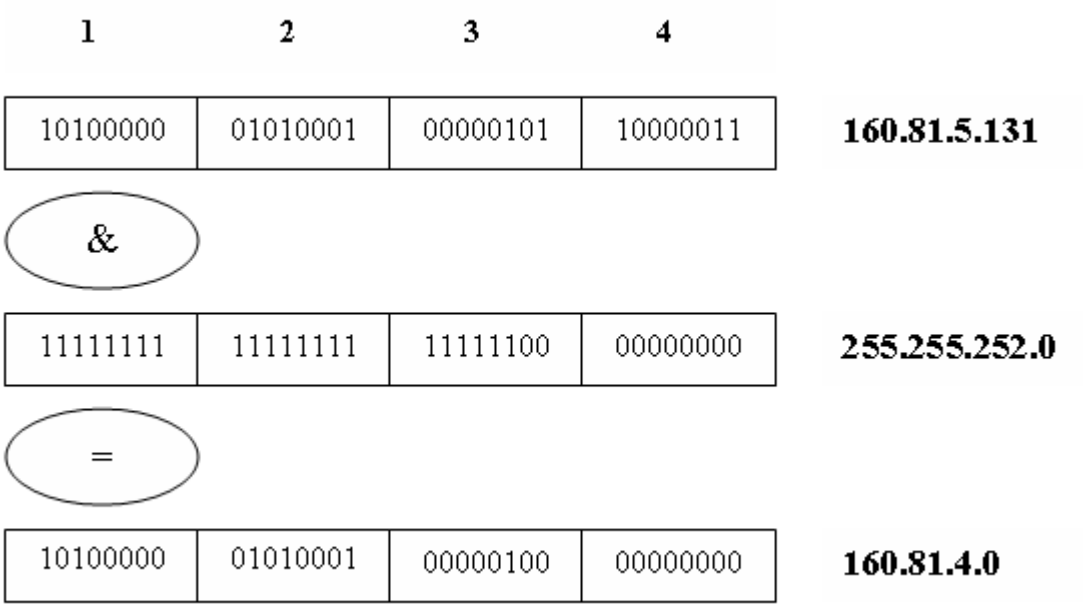

Рисунок 2.4.4. Вычисление адреса сети с помощью маски сети

Протокол IP обеспечивает доставку дейтаграмм в пределах всей ***составной*** IP-сети. Составной IP-сетью называются объединение нескольких IP-сетей с помощью специальных устройств, называемых ***шлюзами***. Обычно шлюз представляет собой компьютер, на котором установлены несколько интерфейсов IP и специальное программное обеспечение, реализующее протоколы Межсетевого уровня. На рисунке 2.4.5 изображен пример составной IP-сети. Шлюз имеет два интерфейса: один принадлежит сети 172.16.5.0 другой сети – сети 172.16.12.0. Обе сети имеют маску 255.255.252.0, что соответствует 22 битовому адресу сети. Для обмена данными с хостом, который находится в другой сети, используется ***таблица***

маршрутов. Таблица маршрутов имеется на каждом ***узле*** сети (хост или шлюз) и содержит информацию об адресах сетей, адресах шлюзов и т.п. Процесс определения адреса следующего узла в пути следования дейтаграммы и пересылка ее по этому адресу называется ***маршрутизацией***. С процессом маршрутизации в IP-сети можно ознакомиться в [5,6].

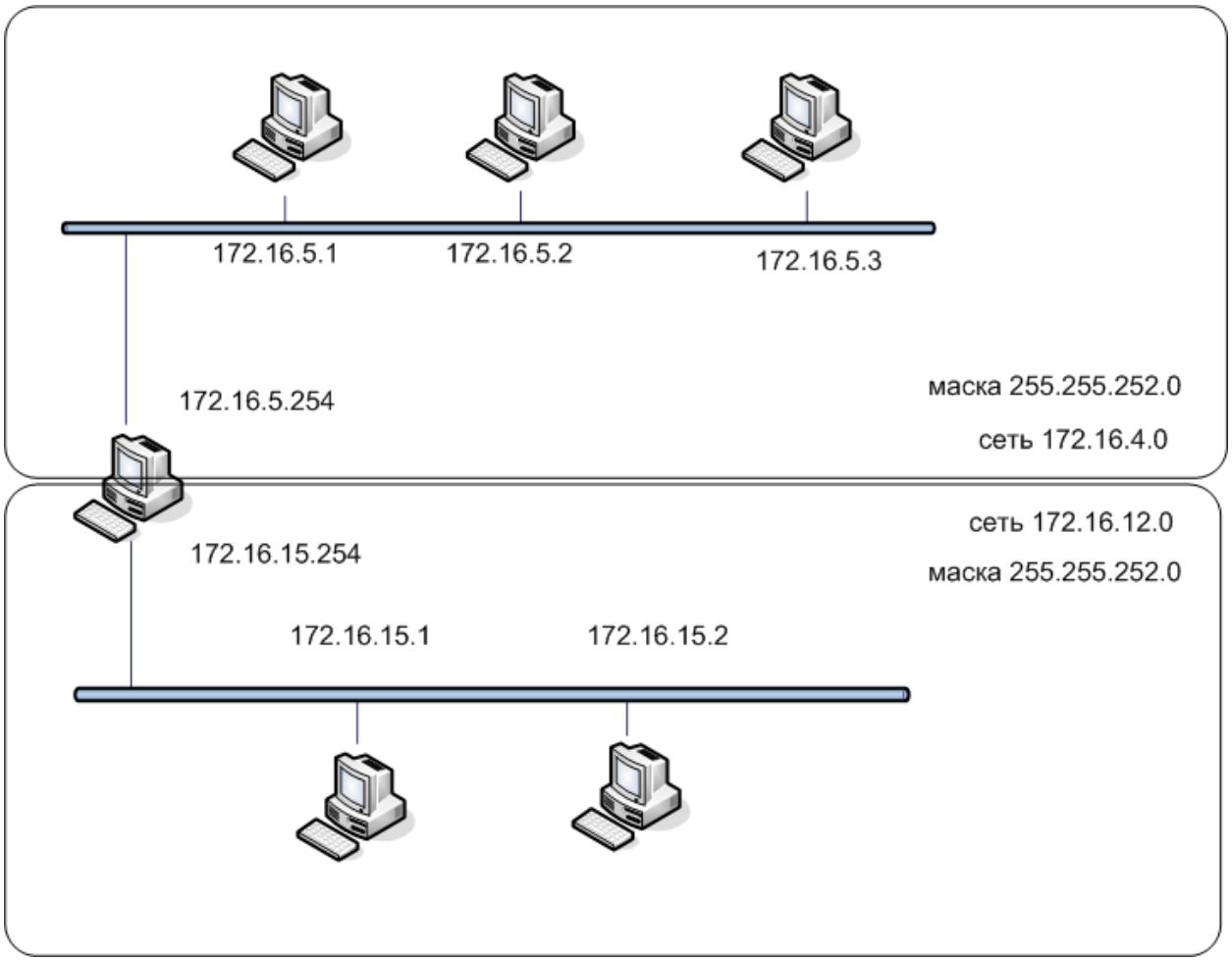

Рисунок 2.4.5. Пример составной IP-сети

Протокол ICMP. Спецификация протокола ICMP (Протокол контроля сообщений в Internet) изложена в документе RFC 792. ICMP является неотъемлемой частью TCP/IP и предназначен для транспортировки информации о сетевой деятельности и маршрутизации. IPCM сообщения представляют собой специально отформатированные IP-дейтаграммы, которым соответствуют определенные типы (15 типов) и коды сообщений. Описание типов и кодов IPCM-сообщений содержится в [5]. С помощью протокола ICMP осуществляется деятельность утилит достижимости (***ping***, ***traceroute***); регулируется частота отправки IP-дейтаграмм, оптимизируется MTU для маршрута передачи IP-дейтаграмм; доставляется хостам, маршрутизаторам и шлюзам всевозможная служебная информация;

осуществляется поиск и переадресация маршрутизаторов; оптимизируются маршруты; диагностируются ошибки и оповещаются узлы IP-сети.

Протокол ARP. IP-адреса могут восприниматься только на сетевом уровне и вышестоящих уровнях TCP/IP. На канальном уровне всегда действует другая схема адресации, которая зависит от используемого протокола. Например, в сетях Ethernet используются 48-битные адреса. Для установки соответствия между 32-разрядными IP-адресами и теми или иными MAC-адресами, действующими на канальном уровне, применяется механизм привязки адресов по протоколу ARP, спецификация которого приведена в документе RFC 826. Основной задачей ARP является динамическая (без вмешательства администратора, пользователя, прикладной программы) проекция IP-адресов в соответствующие MAC-адреса аппаратных средств. Эффективность работы ARP обеспечивается тем, что каждый хост кэширует специальную ARP-таблицу. Время существования записи в этой таблице составляет обычно 10- 20 минут с момента ее создания и может быть изменено с помощью параметров реестра [6]. Просмотреть текущее состояние ARP-таблицы можно с помощью команда ***arp***. Кроме того, протокол ARP используется для проверки существования в сети дублированного IP-адреса и разрешения запроса о собственном MAC-адресе хоста во время начальной загрузки.

Протокол RARP. Протокол RARP, как следует из названия (Reverse ARP), по своей функции противоположен протоколу ARP. RARP применятся для получения IP-адреса по MAC-адресу. В настоящее время протокол заменен на протокол Прикладного уровня DHCP, предлагающий более гибкий метод присвоение адресов.

Другие протоколы Межсетевого уровня. Следует отметить, что на Межсетевом уровне TCP/IP могут использоваться и другие протоколы: ***RIP*** (RFC 1058) – основной дистанционно-векторный протокол маршрутизации; ***OSPF*** (RFC 2328) – протокол первоначального открытия кратчайших маршрутов; ***BGP*** (RFC 1771) – пограничный межсетевой протокол и т.д. Сведения о назначении этих протоколов описаны в [5.6].

Протокол IPv6. Наиболее распространенной на настоящий момент версией протокола IP является IPv4 – именно об этой версии говорилось выше. Этот протокол оказался самым удачным сетевым протоколом из всех, когда-либо созданных. Поэтому IPv4 быстро превратился в стандарт. Можно сказать, что протокол IPv4 стал жертвой собственной популярности, т.к. предлагаемое полезное пространство адресов практически исчерпано. Как результат усилий направленных на решение этой проблемы появился протокол IPv6, в котором попутно было реализовано много других новых возможностей. Главным отличительным признаком протокола IPv6 является 128-битный адрес, позволяющий увеличить адресное пространство боле чем на 20 порядков. Основная концепция IPv6: каждый отдельный узел должен иметь собственный уникальный идентификатор интерфейса. Кроме того, протокол IPv6 требует соответствие идентификаторов интерфейсов формату IEEE EUI-64, позволяющему применять фиксированные (“зашитые” при

изготовлении в специальную память сетевой платы) MAC-адреса сетевых плат. Например, 48-битный MAC-адрес платы Ethernet изначально предназначен для глобальной идентификации. Первые 24 бита этого адреса обозначают производителя платы (в соответствии с кодировкой ICANN) и индивидуальную партию изделия, а остальные 24 бита определяются производителем, с таким расчетом, чтобы каждый номер был уникален в пределах всей его продукции. Таким образом уникальный идентификатор интерфейса IPv6 на основе Ethernet содержит в младших 64-х разрядах 128-битного адреса MAC-адрес платы Ethernet. Причем дополнение 48-бит адреса MAC-адреса до 64 бит осуществляется добавлением 16 бит (0xFFFF) между двумя его половинами. На первоначальном этапе внедрения IPv6 предполагается совместное использование обеих версий IP-протокола. При этом предполагается использование, так называемых, IPv4-совместимых и IPv4-преобразованных адресов. Другой интересной особенностью IPv6 является возможность ***автоконфигурации***. Автокофигурация – это процесс, позволяющий хосту находить информацию для настройки собственных IP-параметров. В версии IPv6 основным средством позволяющим выполнять подобную настройку является протокол DHCP. Пересмотр процесса автоконфигурации вызван сложность администрирования сетей с большим количеством хостов и в связи с необходимостью поддерживать мобильных (перемещающихся) пользователей. Большое внимание в новой версии протокола уделяется вопросам безопасности Более подробно о перспективном протоколе IPv6 можно узнать из [6].

2.5. Протоколы Транспортного уровня

Основным назначением протоколов транспортного уровня является сквозная доставка данных произвольного размера по сети между прикладными процессами, запущенными на узлах сети. Транспортный уровень TCP/IP представлен двумя протоколами: ***TCP*** (Transmission Control Protocol), чье имя присутствует в названии всего стека; **UDP** (User Datagram Protocol) – протокол передачи дейтаграмм пользователя. Процесс, получающий или отправляющий данные с помощью Транспортного уровня, идентифицируется номером, который называется ***номером порта***. Таким образом, адресат в сети TCP/IP полностью определяется тройкой: IP-адресом, номером порта и типом протокола транспортного уровня (UDP или TCP). Основным отличием протоколов UDP и TCP является, то, что UDP – протокол без установления соединения (ориентированным на сообщения), а TCP – протокол на основе соединения (ориентированный на поток). На рисунке 2.5.1 изображен стек протоколов TCP в двух разрезах и отображены названия принятые для обозначения блоков данных в протоколах TCP и UDP. Движение информации с верхних уровней на нижние сопровождается ***инкапсуляцией данных*** (упаковкой по принципу матрешки), а движение в обратном направлении – распаковкой. Отправляемый кадр данных содержит в себе всю необходимую информацию, чтобы быть доставленным (на

Уровне доступа к сети) и правильно распакованным на каждом уровне получателя и, наконец, предоставленным на Прикладном уровне в виде, пригодным для использования процессом.

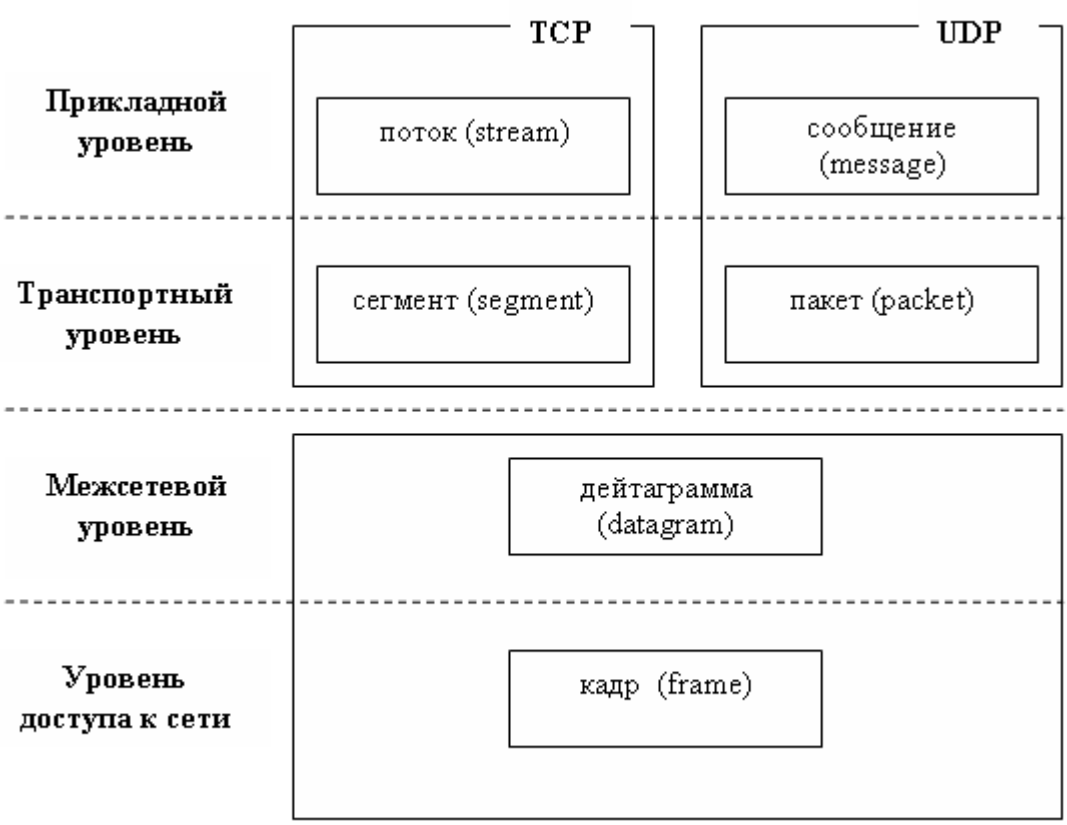

Рисунок 2.5.1. Протоколы TCP и UDP в стеке TCP/IP

Номера портов, используемые для идентификации прикладных процессов (в соответствии с документами IANA), делятся на три диапазона: ***хорошо известные номера портов (well-known port number)***, ***зарегистрированные номера портов (registered port number)***, ***динамически номера портов (dynamic port number)***. Распределение номеров портов по диапазонам приведено в таблице 2.5.1.

Таблица 2.5.1

Хорошо известные номера портов	0 – 1023
Зарегистрированные номера портов	1024 – 49151
Динамические номера портов	49152– 65535

Хорошо известные номера портов присваиваются ***базовым системным службам (core services)***, имеющие системные привилегии. Зарегистрированные номера портов присваиваются промышленным приложениям и процессам. Распределение некоторых хорошо известных и зарегистрированных номеров портов приведено в таблице 2.5.2. Динамические номера портов (их часто называют ***эфемерными портами***) выделяются, как правило, прикладным процессам специализированной службой операционной системы. Некоторые системы TCP/IP применяют

диапазон значений от 1024 до 5000 для назначения эфемерных номеров портов.

Таблица 2.5.2

Номер порта	Протокол	Описание
20	TCP	File Transport Protocol (FTP)
21	TCP	File Transport Protocol (FTP)
22	TCP	Secure Shell (SSH)
23	TCP	Telnet
25	TCP	Simple Mail Transfer Protocol (SMTP)
53	TCP	Domain Name Server (DNS)
53	UDP	Domain Name Server (DNS)
66	TCP	Oracle SQL*NET
67	UDP	Dynamic Host Configuration Protocol (DHCP) Server
68	UDP	Dynamic Host Configuration Protocol (DHCP) Client
80	TCP	World Wide Web (WWW)
110	TCP	Post Office Protocol, Version 3 (POP3)
111	TCP	Remote Procedure Call (RPC)
111	UDP	Remote Procedure Call (RPC)
143	TCP	Internet Message Access Protocol (IMAP4)
1352	TCP	Lotus Notes
1433	TCP	Microsoft SQL Server
1522	TCP	Oracle SQL Sever

Протокол UDP. Протокол UDP является протоколом без установления соединения. Спецификация протокола описывается в документе RFC 768. Основными свойствами протокола являются:

1) отсутствие механизмов обеспечения надежности: пакеты не упорядочиваются, и их прием не подтверждается;
2) отсутствие гарантий доставки: пакеты оправляются без гарантии доставки, поэтому процесс Прикладного уровня (программа пользователя) должен сам отслеживать и обеспечивать (если это необходимо повторную передачу);
3) отсутствие обработки соединений: каждый оправляемый или получаемый пакет является независимой единицей работы; UDP не имеет методов установления, управления и завершения соединения между отправителем и получателем данных;
4) UDP может по требованию вычислять контрольную сумму для пакета данных, но проверка соответствия контрольной сумы ложится на процесс Прикладного уровня;
5) отсутствие буферизации: UDP оперирует только одним пакетом и вся работа по буферизации ложится на процесс Прикладного уровня;

6) UDP не содержит средств, позволяющих разбивать сообщение на несколько пакетов (фрагментировать) – вся эта работа возложена на процесс Прикладного уровня.

Следует обратить внимания, что протокол UDP характеризуется тем, что он не обеспечивает. Все перечисленные отсутствующие характеристики присутствуют в протоколе TCP. Фактически UDP – это тонкая прослойка интерфейса, обеспечивающая доступ процессов Прикладного уровня непосредственно к протоколу IP.

Протокол TCP. Протокол TCP является ***надежным*** байт-ориентированным протоколом с ***установлением соединения***. При получении дейтаграммы, в поле Protocol (со структурой IP-дейтаграммы можно ознакомиться в [5,6]) которой указан код 6 (код протокола TCP) IP-протокол извлекает из дейтаграммы данные, предназначенные для Транспортного уровня, и переправляет их модулю протокола TCP. Модуль TCP анализирует служебную информацию заголовка сегмента (структура TCP-сегмента приведена в [5,6]), проверяет целостность (по контрольной сумме) и порядок прихода данных, а также подтверждает их прием отправляющей стороне. По мере получения правильной последовательности неискаженных данных процесса отправителя, используя поле Destination Port Number заголовка сегмента, модуль TCP переправляет эти данные процессу получателя.

Протокол TCP рассматривает данные отправителя как непрерывный не интерпретируемый (не содержащий управляющих для TCP команд) поток октетов. При этом TCP при отправке разделяет (если это необходимо) этот поток на части (TCP-сегменты) и объединяет полученные от протокола IP-дейтаграммы при приеме данных. Немедленную отправку данных может быть затребовано процессом с помощью специальной функции PUCH, иначе TCP сам решает, когда отправлять данные отправителя и когда их передавать получателю.

Модуль TCP обеспечивает защиту от повреждения, потери, дублирования и нарушения очередности получения данных. Для выполнения этих задач все октеты в потоке данных пронумерованы в возрастающем порядке. Заголовок каждого сегмента содержит число октетов и порядковый номер первого октета данных в данном сегменте. Каждый сегмент данных сопровождается контрольной суммой, позволяющей обнаружить повреждение данных. При отправлении некоторого числа последовательных октетов данных, отправитель ожидает подтверждение приема. Если подтверждения не приходит, то предполагается, что группа октетов не дошла по назначению или была повреждена – в этом случае предпринимается повторная попытка переслать данные.

Протокол TCP обеспечивает одновременно нескольких соединений. Поэтому говорят о ***разделении каналов***. Каждый процесс Прикладного уровня идентифицируется номером порта. Заголовок TCP-сегмента содержит номера портов отправителя и получателя.

На рисунке 2.5.2 прерывистыми линиями изображены каналы между процессами Прикладного уровня A, B, C, D и E. Процесс D работает на хосте с IP-адресом 172.16.5.2 (рисунок 2.5.1) и использует порты 2777 и 2888 для связи с двумя процессами A (порт 2000) и B (порт 2500), функционирующими на хосте с IP-адресом 172.16.15.1. Процессы C и E образуют канал между хостами 172.16.5.1 и 172.16.5.3 и используют порты 2500 и 2000 соответственно.

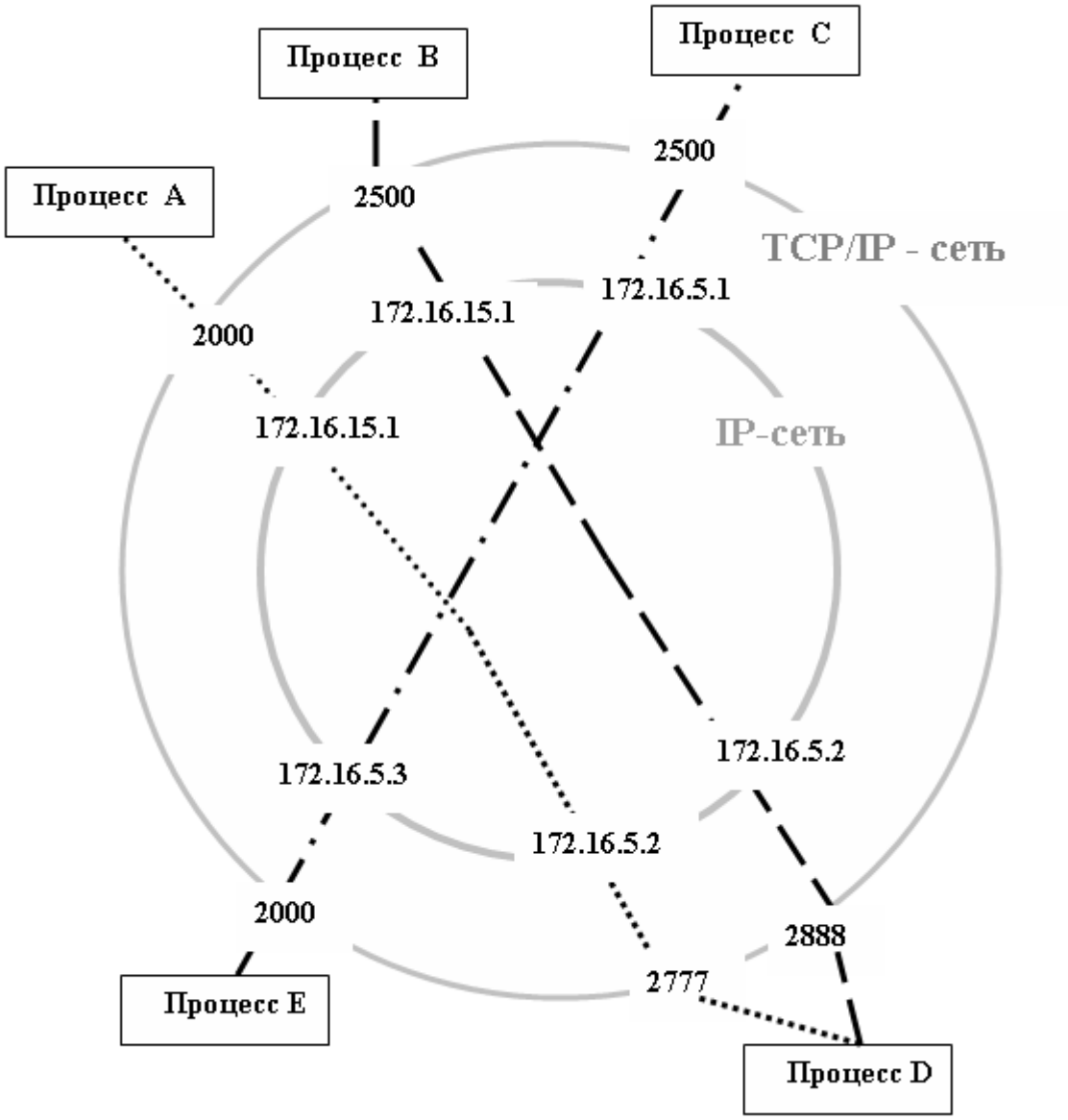

Рисунок 2.5.2. Разделение каналов в сети TCP/IP

Совокупность IP-адреса и номера порта называется ***сокетом***. Сокет однозначно идентифицирует прикладной процесс в сети TCP/IP. Следует помнить, что одни и те же номера портов могут быть использованы как для протокола UDP, так и для протокола TCP.

2.6. Интерфейс внутренней петли

Большинство реализаций TCP/IP поддерживает ***интерфейс внутренней петли (loopback interface)***, который позволяет двум прикладным процессам, находящимся на одном хосте, обмениваться данными посредством протокола TCP/IP. При этом, как обычно, формируются дейтаграммы, но они не покидают пределы одного хоста. Для интерфейса внутренней петли, как уже упоминалось выше, зарезервирована сеть 127.0.0.0. В соответствии с общепринятыми соглашениями, большинство операционных систем назначают для интерфейса внутренней петли адрес 127.0.0.1 и присваивают символическое имя ***localhost***.

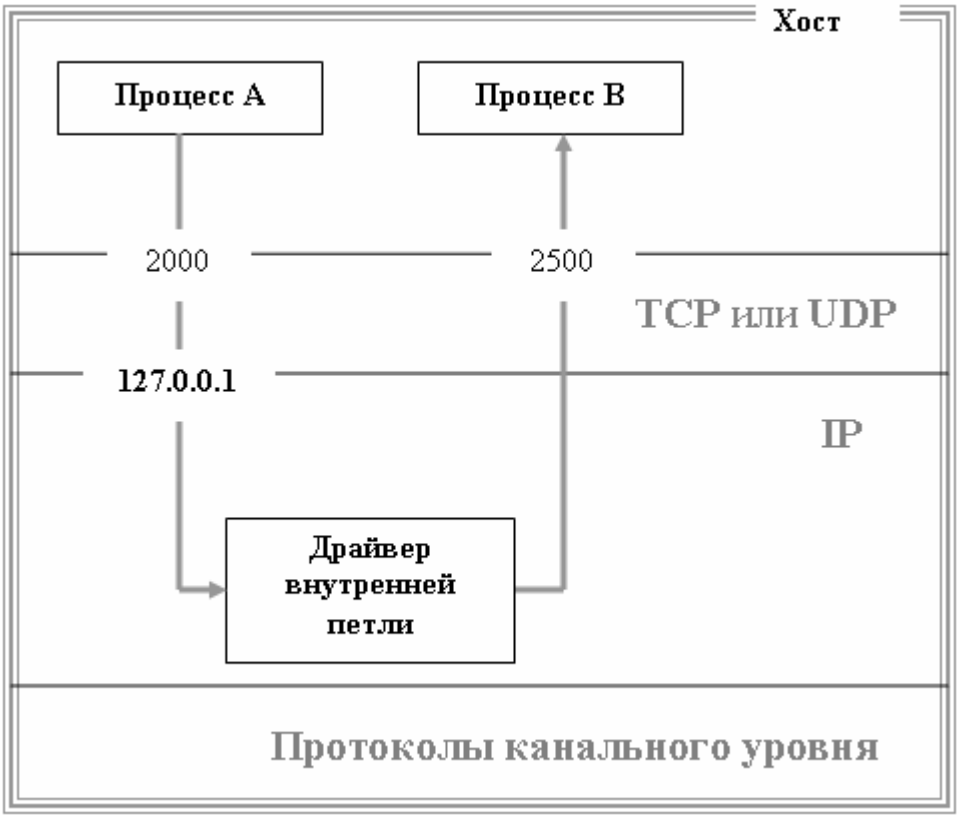

Рисунок 2.6.1. Схема работы интерфейса внутренней петли

На рисунке 2.6.1 приведена упрощенная схема обработки данных интерфейсом внутренней петли. Прикладной процесс А, изображенный на рисунке, используя номер порта 2000, отправляет данные процессу В. Указав в параметрах сокета процесса В сетевой адрес 127.0.0.1, процесс А обеспечил обработку посылаемых дейтаграмм на Межсетевом уровне драйвером внутренней петли, который направляет эти дейтаграммы во входную очередь модуля IP. IP- модуль, следуя обыкновенной логике своей работы, доставляет данные на Транспортный уровень. Далее протокол

Транспортного уровня в соответствии с номером порта 2500 в заголовке сегмента (или пакета) направляет данные процессу B.

Следует обратить внимание на следующее: все данные пересылаемые по интерфейсу внутренней петли не только не покидают пределов хоста, но и не затрагивают никаких внешних механизмов за пределами стека TCP/IP.

2.7. Интерфейсы сокетов и RPC

Для предоставления возможности разработчикам программного обеспечения использовать процедуры стека TCP/IP в состав операционных систем включаются специальные интерфейсы (***Application Programming Interface, API***), представляющие собой, как правило, набор специальных функций и технологических инструкций, обеспечивающих доступ к модулям протокола TCP/IP.

Наиболее распространенными API для обмена данными в сети, являются ***интерфейс сокетов*** и ***RPC (Remote Procedure Call)*** – вызов удаленных процедур.

API сокетов. API сокетов – это название программного интерфейса, предназначенного для обмена данными между процессами, находящимися на одном или на разных объединенных сетью компьютерах. Сокетом, кроме того, называют абстрактный объект, представляющий оконечную точку соединения. Впервые этот интерфейс появился в 1980-х годах в операционной системе BSD Unix (Berkeley Software Distribution), разработанной в университете Беркли (США, Калифорния) и описан в стандарте ***POSIX*** (Portable Operating System Interface for Unix).

Стандарт POSIX – это набор документов, описывающих интерфейсы между прикладной программой и операционной системой. Стандарт создан для обеспечения совместимости различных Unix-подобных операционных систем и переносимости исходных программ на уровне исходного кода. Официально стандарт определен как IEEE 1003, международное название стандарта ISO/IEC 9945.

В той или иной мере интерфейс сокетов поддерживается большинством современных операционных систем. Это дает возможность, например, Unix-процессу, реализованному на языке C с использованием API сокетов, обмениваться данными с Windows-приложением или с приложением, работающим на мэйнфрейме, если эти приложения используют API сокетов. Например, в операционной системе Windows интерфейс сокетов имеет название ***Windows Sockets API***. API сокетов включает в себя функции создания сокета (имеется в виду объект операционной системы, описывающий соединение), установки параметров сокета (сетевой адрес, номер порта и т.д.), функции создания канала и обмена данными между сокетами. Кроме того, есть набор функций позволяющий управлять передачей данных, синхронизировать процессы передачи и приема данных, обрабатывать ошибки и т.п. Следует отметить, что интерфейсы сокетов, поддерживаемые различными операционными

системами, отличаются друг от друга, но все обеспечивают работу сокетов в стандартном режиме. В этом случае говорят о ***BSD-сокетах.***

Интерфейс сокетов используется большинством программных систем имеющих архитектуру клиент-сервер. К ним относятся сетевые службы, Web-серверы, серверы баз банных, серверы приложений и т.п.

Интерфейс RPC. Интерфейс RPC определяет программный механизм, который первоначально был разработан в компании Sun Microsystems и предназначался для того, чтобы упростить разработку распределенных приложений. Спецификация RPC компании Sun Microsystems содержится в документах RFC 1059, 1057, 1257.

RPC Sun Microsystems реализована в двух модификациях: одна выполнена на основе API сокетов для работы над TCP и UDP, другая, названная ***TI-RPC (Transport Independent RPC)***, использует API TLI (Transport Layer Interface, компании AT&T) и способна работать с любым транспортным протоколом, поддерживаемый ядром операционной системы.

Идея, положенная в основу RPC, заключается в разработке специального API, позволяющего осуществлять вызов ***удаленной процедуры*** (процедуры, которая находится и исполняется на другом хосте) способом, по возможности, ничем не отличающимся от вызова локальной процедуры из динамической библиотеки. Реализация этой идеи осложняется необходимостью учитывать возможность различия операционных сред, в которых работают вызывающая и вызываемая процедуры (отсюда, различные типы данных, невозможность обрабатывать адресные указатели и т.п.). Кроме того, следует предусмотреть обработку внепланового завершения процедуры на одной из сторон распределенного приложения. Все эти проблемы сделали интерфейс RPC достаточно сложным. Прозрачность механизма вызова достигается созданием вместо вызываемой и вызывающей процедур специальных программных заглушек, называемых ***клиентским*** и ***серверным стабами.***

Клиентским стабом называется тот стаб, который находится на хосте с вызывающей процедурой. Его основной задачей является преобразовать передаваемые параметры в формат стандарта ***XDR (External Data Representation)*** и скрыть (подменив вызываемую удаленную процедуру локальным вызовом стаба) от пользователя механизм RPC.

Серверный стаб находится на том же хосте, что и вызываемая процедура и предназначен для преобразования полученных параметров из формата XDR в формат, воспринимаемый вызываемой процедуры, а также для сокрытия (серверный стаб подменяет вызывающую процедуру на стороне сервера) RPC-механизма от вызываемой процедуры.

Стандарт XDR предназначен кодирования полей в запросах и ответах интерфейса RPC. Стандарт регламентирует все типы данных и уточняет способ их передачи в RPC-сообщениях. Спецификация стандарта XDR приведена в RFC 1014.

Число вызываемых удаленных процедур не регламентируется спецификацией RPC. Поэтому за ними не закрепляются конкретные TCP-

порты. Порты получают сами удаленные процедуры динамическим образом (эфемерные порты). Учет соответствия портов вызываемым процедурам осуществляет специальная программа ***PortMapper*** (регистратор портов). Сам PortMapper доступен по 111 порту и тоже является удаленной процедурой. Процедура PortMapper – это связующее звено между различными компонентами системы. Всякая вызываемая процедура, должна быть зарегистрирована в базе данных PortMapper с помощью специальных служебных функций. Вызывающая сторона (клиентский стаб) с помощью все тех же служебных функций может получить спецификацию вызываемой процедуры.

Развитием технологии RPC для объектно-ориентированного программирования в операционной системе Windows являются технологии ***COM*** и ***DCOM***, которые позволяют создавать удаленные объекты.

Аналогом RPC в Java-технологиях является механизм ***RMI (Remote Method Invocation)***, позволяющий работать с удаленными Java-объектами. Информацию об объекте и его методах вызывающая сторона может получить, обратившись к реестру RMI (аналогу PortMapper).

Сетевая файловая система ***NFS (Network File System)***, повсеместно используемая в качестве службы прозрачного удаленного доступа к файлам основана на механизме RPC Sun Microsystems.

Следует отметить, что кроме Sun-реализации интерфейса RPC, широко применяется и конкурирующий программный продукт, разработанный объединением OSF (Open Software Foundation).

2.8. Основные службы TCP/IP

Программную реализацию Протоколов Прикладного уровня TCP/IP принято называть службами. Работа служб TCP/IP определяется множеством соглашений: спецификациями структур сообщений, поддерживаемых данной службой; регистрацией хорошо известного порта используемого службой; спецификациями программных компонентов, необходимых для работы службы и т.д.

Как правило, служба реализуются в виде сервера, предоставляющего услуги клиентам (другим процессам). В этом случае клиенты использует запросы (определенные спецификациями) и получает соответствующий сервис, предусмотренный данному запросу. Однако ярко выраженной архитектурой клиент-сервер обладают не все службы. В некоторых случаях сами службы могут выступать в роли клиента или осуществлять межсерверный (между однородными серверами) обмен данными, обычно используемый для синхронизации (***репликации***) серверов.

Существуют тысячи служб TCP/IP, спецификации которых изложены в документах RFC. Здесь рассматриваются лишь те, которые принято называть ***традиционными службами TCP/IP***.

2.8.1. Служба и протокол DNS

Служба ***DNS (Domain Name System)*** является одной из важнейших служб TCP/IP, само появление которой в 1980-х годах дало мощный толчок развитию TCP/IP и всемирной сети Internet. Дело в том, что DNS обеспечивает важную возможность преобразования символических доменных имен в соответствующие IP-адреса (***разрешение имен***). Например, для обращения к адресу серверу компании Microsoft, имеющему IP-адрес 207.46.230.229, можно обратиться, используя символическое имя microsoft.com. С одной стороны, это дает более наглядную нотацию, а с другой, появляется возможность не привязывать жестко получение услуг сервера к фиксированному адресу, который при реорганизации сети может измениться.

Службу DNS можно рассматривать, как распределенную иерархическую базу данных, основное назначение которой отвечать на два вида запросов: выдать IP-адрес по символическому имени хоста и наоборот – выдать символическое имя хоста по его IP-адресу. Облуживание этих запросов и поддержку базы данных в актуальном состоянии обеспечивают взаимодействующие глобально рассредоточенные в сети Internet серверы DNS. База данных имеет древовидную структуру, в корне которой нет ничего, а сразу под корнем находятся ***первичные*** сегменты ***(домены)***: **.com, .edu, .gov, …, .ru, .by, .uk, …** Наименование этих первичных доменов отражает деление базы данных DNS по отраслевому (домены, обозначенные трехбуквенным кодом) и национальному признакам (двухбуквенные домены в соответствии со стандартом ISO 3166). ***Доменом*** в терминологии DNS называется любое поддерево дерева базы данных DNS.

DNS-серверы, обеспечивающие работоспособность всей глобальной службы, тоже имеют древовидную структуру подчиненности, которая соответствует структуре распределенной базы данных. По своему функциональному назначению DNS-серверы бывают: первичные серверы (которые являются главными серверами, поддерживающими свою часть базы данных DNS), вторичные серверы (всегда привязан к некоторому первичному серверу и используются для дублирования данных первичного сервера), кэширующие серверы (обеспечивают хранение недавно используемых записей из других доменов и служат для увеличения скорости обработки запросов на разрешение имен).

При обработке запроса на разрешение имени, хост, как правило, обращается к первичному или вторичному DNS-серверу, обслуживающему данный домен сети. В зависимости от сложности запроса, DNS-сервер может сам ответить на запрос или переадресовать к другому серверу DNS. Последней инстанцией в разрешении имен являются пятнадцать (на сегодняшний день) корневых серверов имен, представляющих собой вершину всемирной иерархии DNS.

Разработчик приложения может обратиться за разрешением имени с помощью функций, имеющих, как правило, имена ***gethostbyname*** и ***gethostbyaddr***.

Более подробно с принципами работой службы DNS можно ознакомиться в [6].

2.8.2. Служба и протокол DHCP

DHCP (Dynamic Host Configuration Protocol) – это сетевая служба (и протокол) Прикладного уровня TCP/IP, обеспечивающая выделение и доставку IP-адресов и сопутствующей конфигурационной информации (маска сети, адрес локального шлюза, адреса серверов DNS и т.п.) хостам. Применение DHCP дает возможность отказаться от фиксированных IP-адресов в зоне действия сервера DHCP. Описание протокола DHCP содержится в документах: RFC 1534, 2131, 2132, 2141.

Конструктивно служба DHCP состоит из трех модулей: ***сервера DHCP (DHCP Server)***, ***клиента DHCP (DHCP Client)*** и ***ретранслятора DHCP (DHCP Replay Agent)*** .

DHCP-серверы способны управлять одним или несколькими диапазонами IP-адресов (***адресными пулами***). В пределах одного пула можно всегда выделить адреса, которые не должны распределяться между хостами. DHCP-серверы используют для приема запросов от DHCP-клиентов порт 67. Выделение IP-адресов может быть трех типов: ***ручной***, ***автоматический*** и ***динамический*** [5,6]. Обычно DHCP-серверы устанавливают на компьютерах, исполняющих роль сервера в сети.

DHCP-клиенты представляет собой программный компонент, обычно реализуемый как часть стека протоколов TCP/IP и предназначен для формирования и пересылки запросов к DHCP-серверу на выделение IP-адреса, продления срока аренды IP-адреса и т.п. DHCP- клиенты используют для приема сообщений от DHCP-сервера порт 68.

Логика работы протокола DHCP достаточно проста. При физическом подключении к сети, хост пытается подсоединиться к сети, используя для этого DHCP-клиент. Для обнаружения DHCP-сервера DHCP-клиент выдает в сеть широковещательный запрос (это процесс называется ***DHCP-поиском***). Если в этом домене есть DHCP-сервер, то он окликается, посылая клиенту специальное сообщение, содержащее IP-адрес DHCP-сервера. Если доступны несколько DHCP-серверов, то, как правило, выбирается первый ответивший. Получив адрес сервера, клиент формирует запрос на выделение IP-адреса из пула адресов DHCP-сервера. В ответ на запрос, DHCP-сервер выделяет адрес клиенту на определенный период времени (***аренда адреса***). После получения IP-адреса TCP/IP-стек клиента начинает его использовать. Продолжительность аренды адреса устанавливается специально или по умолчанию (может колебаться от нескольких часов до нескольких недель). После истечения срока аренды DHCP-клиент пытается снова договорится с DHCP-сервером о продлении срока аренды или о выделении нового IP-адреса.

Ретранслятор DHCP используется в том случае, если на первоначальном этапе подключения к сети широковещательные запросы DHCP-клиента не могут быть доставлены (по разным причинам) DHCP-

серверу. Ретранслятор в этом случае играет роль посредника между DHCP-клиентом и DHCP-сервером.

Протокол DHCPv6. Протокол DHCPv6 – это новый протокол, работающий над IPv6. Основные задачи DHCPv6 не сильно отличаются от задач выполняемых его предшественником – протоколом DHCPv4 (выше он назывался просто DHCP). Помимо очевидного отличия в длине и формате IP-адресов, значительное расхождение заключается в том, что IPv6-узлы теперь смогут получить (хотя бы локально функционирующие) IPv6- адреса без помощи DHCPv6. Таким образом, осуществляя поиск DHCPv6-сервера IPv6-узлы уже обладают некоторым IPv6-адресом. По-все видимости, основным назначением DHCPv6-протокола будет выделение глобально уникальных IPv6-адресов для тех интерфейсов, которые не могут их сформировать сами и для пересылки дополнительной конфигурационной информации IPv6-хостам.

2.8.3. NetBIOS over TCP/IP

Система ***NetBIOS (Network Basis Input/Output System)*** была разработана в 1985 году в компании Sytek, а позже была заимствована IBM и Microsoft как средство присвоение имен сетевым ресурсам в небольших одноранговых сетях. Вначале NetBIOS была не протоколом, а программным интерфейсом (API) для обращения к сетевым ресурсам. В качестве транспортного протокола выступал ***NetBEUI (NetBIOS Enhanced User Interface)*** – расширенный пользовательский интерфейс NetBIOS.

Служба NetBIOS, применяющая TCP/IP в качестве транспортного протокола (NetBIOS over TCP/IP) называется ***NetBT*** или ***NBT***. Служба NBT, по отношению к стандартному NetBIOS, претерпела значительные изменения, позволяющие передавать NBT-имена компьютеров и команды NBT по TCP/IP-соединению. Эти решение были опубликованы в 1987 году в документах RFC 1001, 1002.

На сегодняшний момент система NBT используется операционной системой Windows для совместного использования сетевых ресурсов и разрешения имен компьютеров. NBT-имя компьютера, это то имя которое задается при инсталляции Windows или может быть установлено (или скорректировано) с помощью программы MyComputer. При этом существует три способа разрешения NBT-имени в сети Windows: запрос к серверу DNS (или аналогичной службе), запрос к локальному сегменту сети (с помощью широковещания) и поиск в локальном списке хоста (файл hosts). Единственный реальный способ разрешения имен в крупномасштабных сетях является DNS-разрешение. Теоретически компьютер в сети Windows может иметь два имени NBT-имя и DNS-имя (следует, правда оговориться, что в случае с DNS именуется не компьютер, а IP-интерфейс).

Более подробно с протоколом NBT можно ознакомиться в [5,6].

2.8.4. Служба и протокол Telnet

Сетевая служба Telnet как протокол описана в двух документах RFC 854 (собственно протокол Telnet, 1985г.) и RFC 855 (дополнительные возможности Telnet). Служба создавалась для подключения пользователей к удаленному компьютеру для выполнения вычислений, работы с базами данных, подготовки документов и т.д. Кроме того, служба применяется для управления работой удаленного компьютера, для настройки и диагностирования сетевого оборудования.

Служба Telnet имеет архитектуру клиент-сервер. Стандартно для обмена данными между клиентом и сервером используется порт 23, но часто используют и другие порты (это допускается протоколом).

В основу службы Telnet положены три фундаментальные идеи: 1) концепция ***виртуального терминала NVT(Network Virtual Terminal)***; 2) принцип договорных опций (согласование параметров взаимодействия); 3) симметрия связи между терминалом и процессом.

Виртуальный терминал, представляет собой спецификацию, позволяющую клиенту и серверу преобразовать передаваемые данные в стандартную (понятную для обеих сторон) форму, позволяющую вводить, принимать и отображать эти данные. Применение виртуального терминала позволяет унифицировать характеристики различных устройств и этим обеспечить их совместимость. В традиционном смысле виртуальный терминал сети понимается как клавиатура печатающего устройства, принимающая и печатающая байты с другого хоста.

Опции представляют собой параметры или соглашения, применяемые в ходе Telnet-соединения. К примеру, опция Echo определяет, отражает ли хост Telnet символы данных, получаемых по соединению. Некоторые опции требуют обмена дополнительной информацией. До обмена дополнительной информацией оба хоста (клиент и сервер) должны согласится на обсуждение параметров, а затем использовать команду SB для начала обсуждения. Полный список опций Telnet опубликован на сайте http://www.iana.org.

Возможность указания TCP-порта при подключении к хосту, позволяет использовать Telnet для диагностирования других Internet-служб.

Серьезным недостатком протокола Telnet является передача данных в открытом виде. Этот недостаток существенно снижает область применения протокола.

Следует отметить, что существует другие программные продукты, подобные Telnet. Например, протокол ***SSH (Secure Shell)*** или свободно распространяемая программа ***PuTTY*** . При разработке этих продуктов был учтен опыт длительной эксплуатации протокола Telnet и исправлены его основные недостатки.

2.8.5. Служба и протокол FTP

Протокол ***FTP (File Transport Protocol)*** описывает методы передачи файлов между хостами сети TCP/IP с использованием транспортного протокола на основе соединений (TCP). Такое же имя FTP носит служба,

реализующая этот протокол. Описание протокола FTP содержится в документе RFC 959 (1985 г.).

Служба FTP имеет архитектуру клиент-сервер (рисунок 2.8.5.1) и содержит три ключевые компоненты: UI(User Interface), PI (Protocol Interpreter) и DTP (Data Transfer Process).

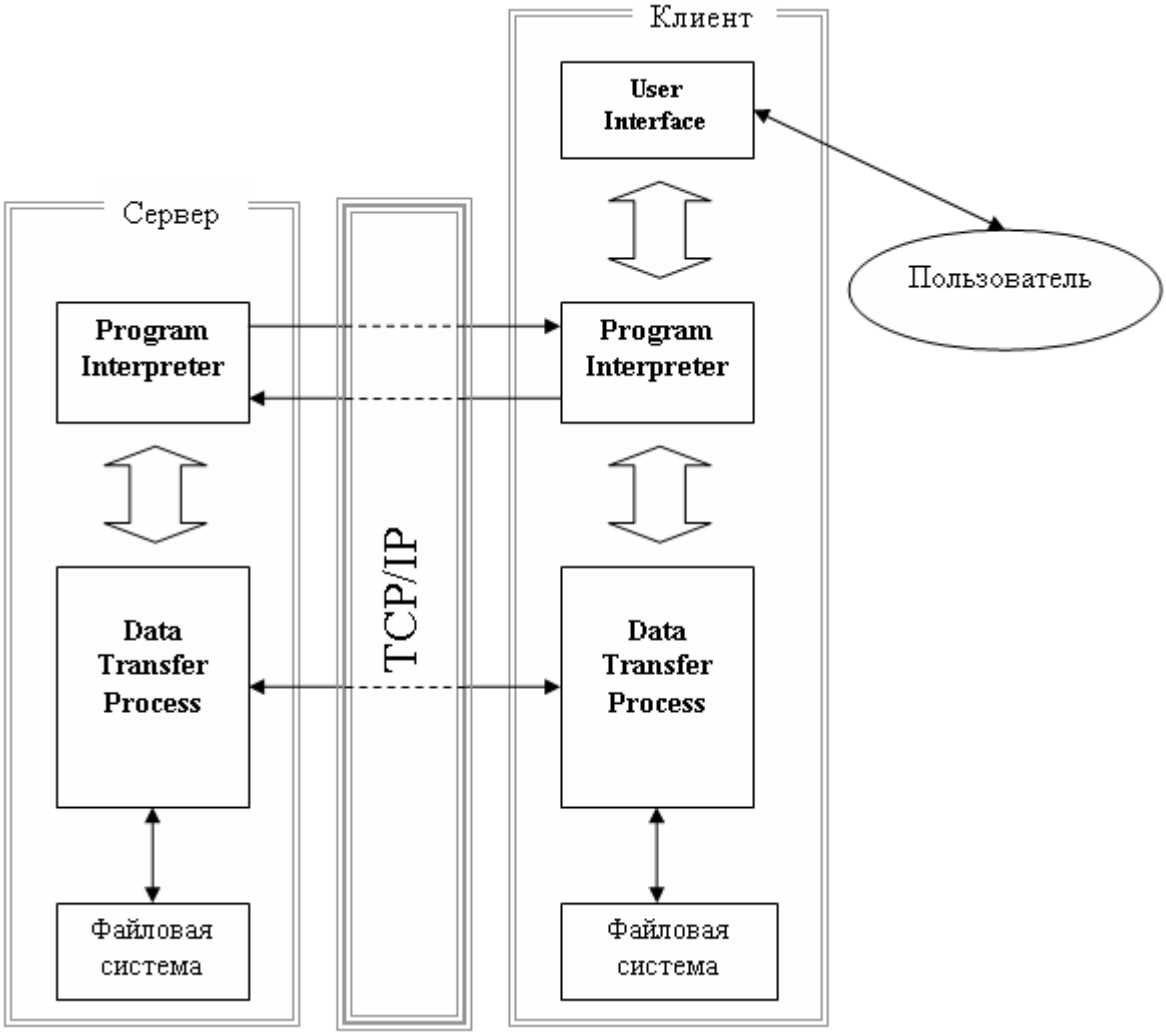

Рисунок 2.8.5.1. Взаимодействие основных компонентов службы FTP

UI – это внешняя оболочка, обеспечивающая интерфейс пользователя. К примеру, клиент FTP операционной системы Windows, обеспечивает интерфейс в виде консоли с командной строкой.

PI – интерпретатор протокола, предназначенный для интерпретации команд пользователя. Кроме того, PI клиента используя эфемерный порт, инициирует соединение с портом 21 PI сервера. Созданный канал (он называется ***каналом управления***) используется для передачи команд пользователя интерпретатору сервера и получения и его откликов. Интерпретатор протокола обрабатывает команды, позволяющие пересылать и удалять файлы, создавать, просматривать и удалять директории и т.п.

DTP – процесс передачи данных, предназначенный для фактического перемещения данных, в соответствии с командами управления, переданными

по каналу управления. Кроме того, на DTP сервера возложена инициатива создания канала передачи данных с DTP клиента. Для этого на стороне клиента используется, как правило, порт 20.

Файловая система на любом конце FTP-соединения может состоять из файлов различного формата: ASCII, EBCDIC, бинарный формат и т.д.

2.8.6. Электронная почта и протоколы SMTP, POP3, IMAP4

Протоколы Прикладного уровня ***SMTP (Simple Mail Transport Protocol)***, ***POP3 (Post Office Protocol)*** и ***IMAP4 (Internet Message Access Protocol)*** являются основой для создания современной электронной почты.

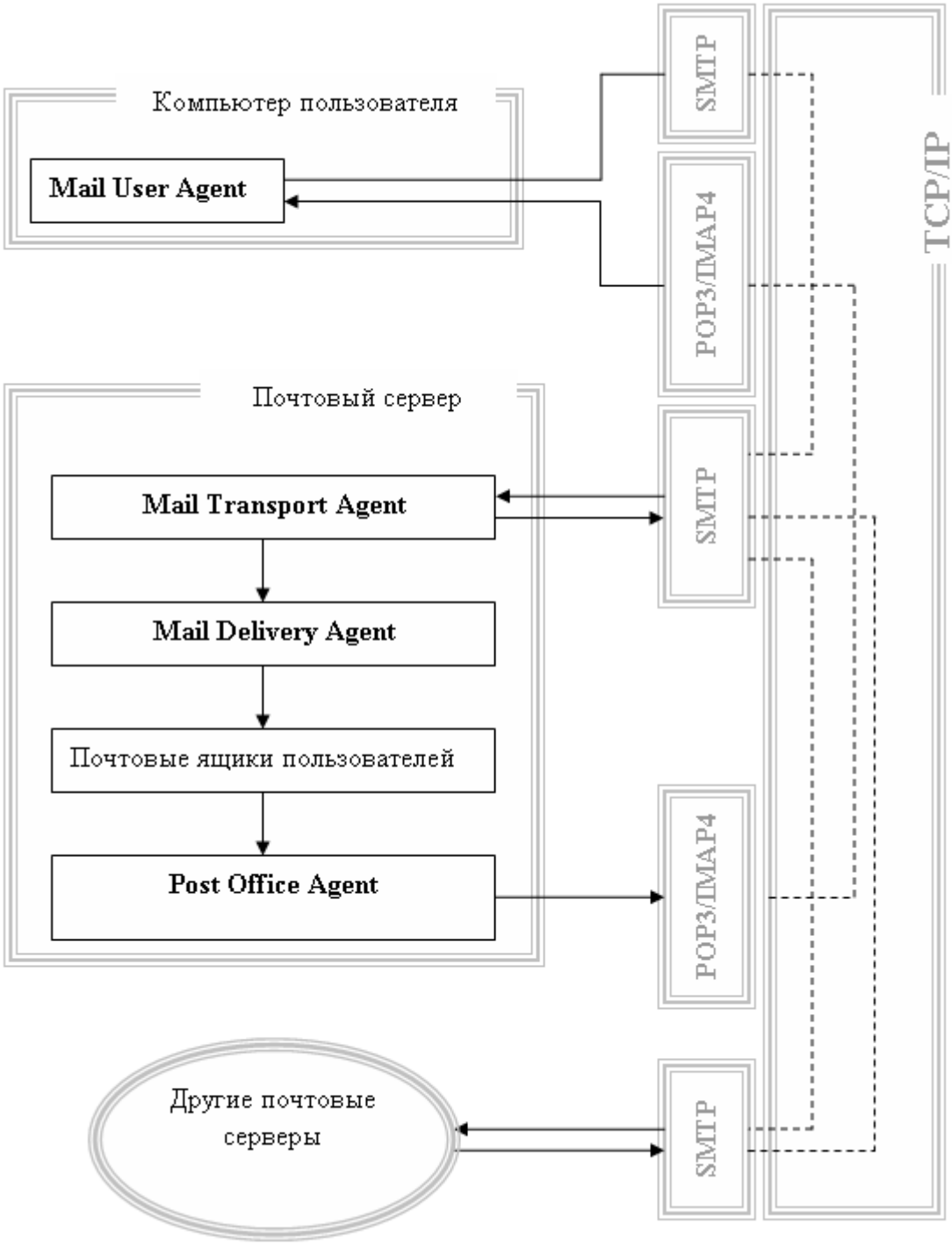

Рисунок 2.8.6.1 Схема взаимодействия компонентов электронной почты

Основными компонентами системы электронной почты являются: ***MTA (Mail Transport Agent)***, ***MDA (Mail Delivery Agent)***, ***POA (Post Office Agent)*** и ***MUA (Mail User Agent)***. На рисунке 2.8.6.1 изображена схема взаимодействия эти компонентов.

MTA – транспортный агент, основное назначение которого: прием почтовых сообщений от пользовательских машин; отправка почтовых сообщений другим MTA (установленных на других почтовых системах); прием сообщений от других MTA; вызов MDA. Это компонент реализован в виде сервера, прослушивающего порт 25 и работающего по протоколу SMTP.

MDA – агент доставки, предназначенный для записи почтового сообщения в почтовый ящик. MDA реализован в виде отдельной программы, которую вызывает MTA по мере необходимости. Обычно, MDA располагают на том же компьютере, что и MTA.

POA – агент почтового отделения, позволяющий пользователю получить почтовое сообщение на свой компьютер. POA реализован в виде сервера, прослушивающего порты 110 и 143. При этом, порт 110 работает по протоколу POP3, порт 143 – IMAP4.

MUA – почтовый агент пользователя позволяет принимать почту по протоколам POP3 и IMAP4 и отправлять почту по протоколу SMTP.

Когда говорят о ***почтовом сервере***, то, обычно подразумевают совокупность серверов MTA, POA, программу MDA, а также систему хранения почтовых сообщений (почтовые ящики) и ряд дополнительных программ, обеспечивающих безопасность и дополнительный сервис, расположенные на отдельном компьютере с TCP/IP-интерфейсом. Наиболее известными являются два почтовых сервера: ***Lotus Notes*** (IBM) и ***Microsoft Exchange Server. Почтовый клиент*** представляет собой программу, устанавливаемую на пользовательском компьютере и взаимодействующую с серверами MTA и POP3, почтового сервера, с помощью TPC/IP – соединения. Например, стандартным клиентом для отправления и организации работы с почтой в ОС Window является программа ***Outlook Express***.

2.8.7. Протокол HTTP и служба WWW

Протокол ***HTTP (Hypertext Transfer Protocol)*** – это протокол Прикладного уровня, доставляющий информацию между различными ***гипермедийными*** системами. Под понятием ***гипермедийной системы*** понимается компьютерное представление системы данных, элементы которой представляются в различных форматах (гипертекст, графические изображения, видеоизображения, звук и т.д.) и обеспечивается автоматическая поддержка смысловых связей между представлениями элементов.

Протокол HTTP применяется в Internet с 1990 года. В настоящее время широкое распространение имеет версия HTTP 1.0, описанная в документе RFC 1945. Разработана новая версия HTTP 1.1 (документ RFC 2616), но пока она находится в стадии предложенного стандарта.

По умолчанию HTTP использует порт 80 и предназначен для построения систем архитектуры клиент-сервер. Запросы клиентов содержат ***URI (Uniform Resource Identifier)*** - универсальный идентификатор ресурса, позволяющий определить у сервера затребованный ресурс. URI представляет собой сочетание ***URL (Uniform Resource Locator)*** и ***URN (Uniform Resource Name)***. URL – унифицированный адресатор ресурсов: предназначен для указания места нахождения ресурса в сети. URN – унифицированное имя ресурса: идентифицирует ресурс, по указанному месту его нахождения (подразумевается, что по данному адресу может быть представлено несколько различных ресурсов). Например, пусть **http://isit301-14:1118/em** – URI, позволяющий вызвать программу Enterprise Manager Oracle Server. Тогда первая часть **http://isit301-14:1118** представляет собой URL (указывает имя хоста и номер порта), а **em** есть URN, идентифицирующее имя ресурса.

Служба ***WWW (World Wide Web)*** предназначена для доступа к ***гипертекстовым документам*** в сети Internet и включает в себя три основных компонента: протокол HTTP, URI-идентификация ресурсов и язык ***HTML (Hyper Text Markup Language)*** .

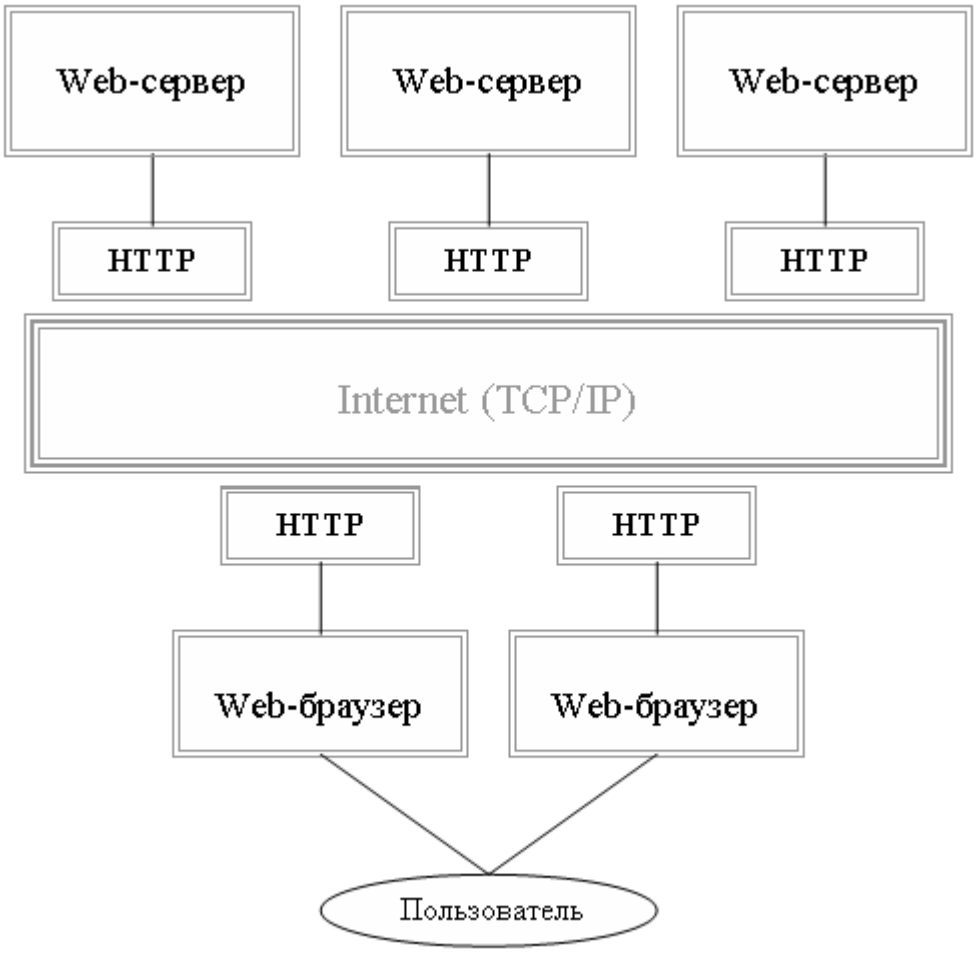

Рисунок 2.8.7.1. Клиент-серверная архитектура службы WWW

HTML – это стандарт оформления гипертекстовых документов. Гипертекстовый документ отличается от любого другого документа тем, что может содержать блоки, физически хранящиеся на разных компьютерах сети Internet. Основной особенностью HTML состоит в том, что форматирование в документе записывается только с помощью ASCII-символов. Одним из ключевых понятий гипертекстового документа является ***гипертекстовая ссылка***. Гипертекстовая ссылка – это объект гипертекстового документа, предназначенный для обозначения URI другого гипертекстового документа.

Как и все службы Internet, служба WWW имеет архитектуру клиент-сервер (рисунок 2.8.7.1). Серверная и клиентская части службы (обычно называемые Web-сервер и Web-браузер) взаимодействуют друг с другом с помощью протокола HTTP. В настоящее время наиболее известными серверными программами являются ***Apache Web Server, Apache Tomcat*** (продукты компании Apache Software Foundation (ASF), распространяемые в соответствии с лицензионным соглашением), ***Microsoft IIS*** (входит в состав дистрибутива последних версий Windows). Наиболее популярными Web-браузерами являются ***Microsoft Internet Explorer***, ***Netscape Navigator***, ***Opera***, ***Mozilla Firebox***.

2.9. Сетевые утилиты

Утилиты представляют собой внешние команды операционной системы и предназначаются для диагностики сети. Сетевые утилиты, поддерживаемые операционной системой Windows, перечислены в таблице 2.9.1. При этом утилиты, наименования которых выделено жирным шрифтом считаются стандартными для протокола TCP/P и присутствуют в большинстве операционных систем.

Таблица 2.9.1

Наименование утилиты	**Назначение утилит**
ping	Проверка соединения с одним или более хостами в сети
tracert	Определение маршрута до пункта назначения
route	Просмотр и модификация таблицы сетевых маршрутов
neststat	Просмотр статистики текущих сетевых TCP/IP-соединений
arp	Просмотр и модификация ARP-таблицы
nslookup	Диагностика DNS-серверов
hostname	Просмотр имени хоста
ipconfig	Просмотр текущей конфигурации сети TCP/IP
nbtstat	Просмотр статистики текущих сетевых NBT-соединений
net	Управление сетью

Утилита ping. Как уже отмечалось раньше, ping в своей работе использует протокол ICMP и предназначена для проверки соединения с удаленным хостом.. Проверка соединения осуществляется путем посылки в

адрес хоста специальных ICMP-пакетов, которые в соответствии с протоколом должны быть возвращены, отправляющему хосту (эхо-пакеты и эхо-ответы).

Для получения справки о параметрах утилиты ping следует выполнить команду ping без параметров. В простейшем случае команда может быть применена с одним параметром:

ping hostname

где hostname – NetBIOS или DNS - имя хоста или его IP-адрес.

Утилит tracert. Как и утилита ping, tracert использует ICMP протокол для определения маршрута до пункта назначения. В результате работы утилиты на консоль выводятся все промежуточные узлы маршрута от исходного хоста до пункта назначения и время их прохождения.

Для получения справки о параметрах утилиты tracert следует выполнить команду tracert без параметров. В простейшем случае команда может быть применена с одним параметром:

tracert hostname

где hostname – NetBIOS или DNS - имя хоста или его IP-адрес.

Утилита route. Утилита route позволяет манипулировать таблицей сетевых маршрутов, которая имеется на каждом компьютере с TCP/IP-интерфейсом. Утилита обеспечивает выполнение четырех команд: print (распечатка таблицы сетевых маршрутов), add (добавить маршрут в таблицу), change (изменение существующего маршрута), delete (удаление маршрута).

Для получения справки о параметрах утилиты route следует выполнить команду route без параметров. В простейшем случае команда может быть использована для распечатки таблицы сетевых маршрутов:

route print

где параметр (команда) print, без уточняющих операндов, указывает на необходимость распечатки всей таблицы.

Утилита netstat. Утилита отражает состояние текущих TCP/IP-соединений хоста, а также статистику работы протоколов. С помощью утилиты netstat можно распечатать номера ожидающих портов всех соединений TCP/IP, имена исполняемых файлов, участвующих в подключениях, идентификаторы соответствующих Windows-процессов и т.д.

Для получения справки о параметрах утилиты netstat, следует выполнить следующую команду.

netstat -?

Активные соединения TCP/IP на компьютере можно просмотреть, набрав на консоли команду nestat с параметром -a.

netstat -a

Утилита arp. Утилита используется для просмотра и модификации ARP-таблицы, используемой для трансляции IP-адресов в адреса протоколов канального уровня (MAC-адреса). С помощью параметров команды можно распечатывать таблицу, удалять и добавлять данные ARP-таблицы. Корректировку ARP-таблицы может осуществлять только пользователь справами администратора.

Для получения справки о параметрах утилиты arp, следует выполнить команду arp без параметров. Получить текущее состояние ARP-таблицы можно с помощью следующей команды.

arp -a

Утилита nslookup. Утилита nslookup предназначена для проверки правильности работы DNS-серверов. С помощью утилиты, пользователь может выполнять запросы к DNS-серверам на получение адреса хоста по его DNS-имени, на получение адресов и имен почтовых серверов, ответственных за доставку почты для отдельных доменов DNS, на получение почтового адреса администратора DNS-сервера и т.д. и т.п. Утилита работает в двух режимах: в режиме однократного выполнения (при запуске в командной строке задается полный набор параметров) и в интерактивном режиме (команды и параметры задаются в режиме диалога). Запуск утилиты в интерактивном режиме осуществляется запуском команды nslookup без параметров.

```
C:\Documents and Settings\Administrator>nslookup
Default Server:  ns1.iptel.by
Address:  80.94.225.5

> www.google.com
Server:  ns1.iptel.by
Address:  80.94.225.5

Non-authoritative answer:
Name:    www.l.google.com
Addresses:  209.85.129.147, 209.85.129.99, 209.85.129.104
Aliases:  www.google.com

> www.oracle.com
Server:  ns1.iptel.by
Address:  80.94.225.5

Non-authoritative answer:
Name:    www.oracle.com
Address:  141.146.8.66

> exit
```

Рисунок 2.9.1. Пример выполнения команды nslookup

На рисунке 2.9.1 демонстрируется работа с утилитой nslookup. Сразу после запуска команды на консоль выводится имя хоста и IP-адрес активного DNS-сервера. После этого в диалоговом режиме были получены IP-адреса хостов с именами www.google.com, www.oracle.com и для завершения работы утилиты была введена команда exit.

Утилита hostname. Утилита предназначена для вывода на консоль имени хоста, на котором выполняется данная команда. Команда hostname не имеет никаких параметров.

Утилита ipconfig. Утилита ipconfig является наиболее востребованной сетевой утилитой. С ее помощью можно определить конфигурацию IP-интерфейса и значения всех сетевых параметров. Особенно эта утилита полезна на компьютерах, работающих с протоколом DHCP: команда позволяет проверить параметры IP-интерфейсов установленные в автоматическом режиме.

Для получения справки о параметрах утилиты следует ввести следующую команду.

```
ipconfig /?
```

Короткий отчет о конфигурации TCP/IP можно получить выдав команду ipconfig без параметров. Для получения полного отчета, можно использовать ключ /all, как это сделано на рисунке 2.9.2.

```
C:\Documents and Settings\Administrator>ipconfig /all

Настройка протокола IP для Windows

        Имя компьютера  . . . . . . . . . : SmeloW
        Основной DNS-суффикс  . . . . . . :
        Тип узла. . . . . . . . . . . . . : гибридный
        IP-маршрутизация включена . . . . : нет
        WINS-прокси включен . . . . . . . : нет

Подключение по локальной сети - Ethernet адаптер:

        DNS-суффикс этого подключения . . :
        Описание  . . . . . . . . . . . . : Broadcom 440x 10/100 Integrated Cont
roller
        Физический адрес. . . . . . . . . : 40-04-29-03-20-24
        Dhcp включен. . . . . . . . . . . : да
        Автонастройка включена  . . . . . : да
        IP-адрес автонастройки. . . . . . : 169.254.152.66
        Маска подсети . . . . . . . . . . : 255.255.0.0
        Основной шлюз . . . . . . . . . . :
        Основной WINS-сервер  . . . . . . : 169.254.152.66
```

Рисунок 2.9.2. Пример выполнения команды ipconfig

Утилита nbtstat. Утилита nbtstat позволяет просматривать статистику текущих соединений, использующих протокол NBT (NetBIOS over TCP/IP). Утилита в чем-то подобна утилите netstat, но применительно к протоколу NBT. Для получения справки о параметрах команды, необходимо ее выполнить без указания параметров.

Утилита net. Утилита net является основным средством управления сетью для сетевого клиента Windows. Команду net часто включают в скрипты регистрации и командные файлы. С помощью этой команды можно зарегистрировать пользователя в рабочей группе Windows, можно осуществить выход из сети, запустить или остановить сетевой сервис, управлять списком имен, пересылать сообщения в сети, синхронизировать время и т.д.

Для вывода списка параметров (команд) утилиты net следует выполнить следующую команду.

```
net help
```

Справка может быть уточнена для каждого отдельного параметра команды. Например, для того, чтобы получить справку для параметра send (пересылка сообщений в сети) следует добавить соответствующий параметр.

```
net help send
```

2.10. Настройка TC/IP в Windows

При установке последних версий операционной системы Windows, поддержка протокола TCP/IP включается автоматически. Работа системного администратора заключается только в настройке параметров сетевого подключения.

Дальнейшее изложение процесса настройки параметров TCP/IP, в основном ориентировано операционную систему Windows XP.

Для настройки параметров в Windows XP, следует использовать окно ***Сетевое подключение*** (Пуск/Панель Управления/Сетевые подключения). Выбрав необходимое сетевое подключение, из предлагаемого списка, получим окно, аналогичное изображенному на рисунке 2.10.1.

Если протокол TCP/IP отсутствует в данном подключении, то установка его может быть выполнена с помощью кнопки **Установить.** Далее будет предложен список компонентов сети для установки.

Если протокол TCP/IP есть, то для его настройки требуется перейти по клавише **Свойства** в окно **Свойства: Internet Protocol (TCP/IP)**. В этом окне требуется ответить на следующие вопросы и заполнить соответчвующие поля.

1. ***Получить IP-адрес автоматически или использовать фиксированный адрес***. Установка признака автоматического получения адреса, подразумевает применение протокола DHCP. Если же адрес фиксируется, то его следует указать в соответствующем поле. Кроме того, в случае фиксированного адреса необходимо установить маску подсети и адрес основного шлюза (шлюз по

умолчанию). В случае автоматического получения IP-адреса, маска подсети и адрес основного шлюза устанавливается по протоколу DHCP.

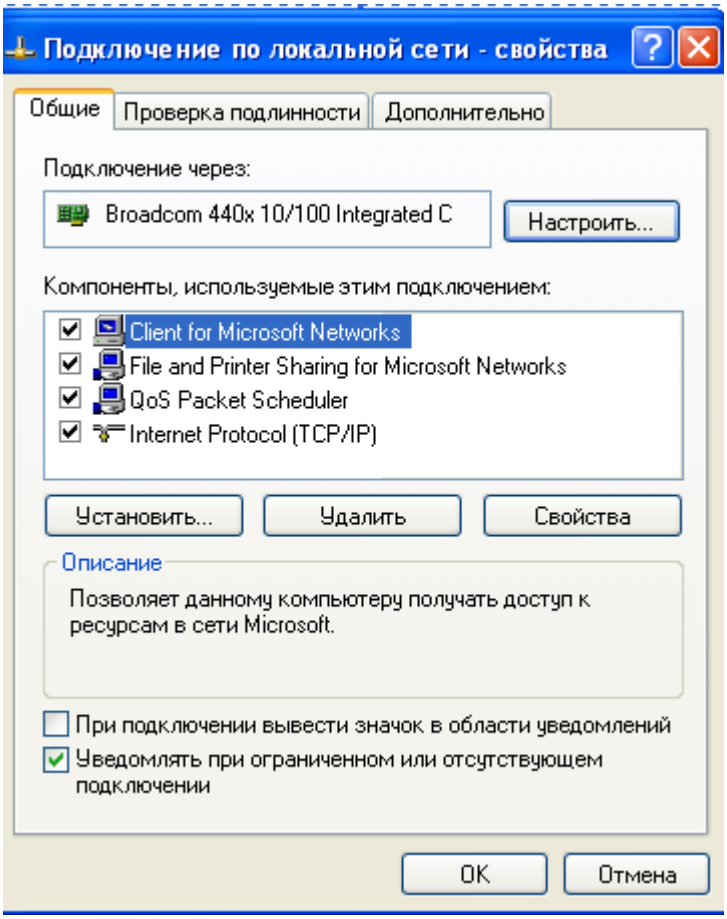

Рисунок 2.10.1 Окно для настройки параметров TCP/IP

2. ***Получить адрес DNS-сервера автоматически или использовать фиксированный адрес.*** Аналогично предыдущему случаю, автоматическая установка подразумевает использования протокола DHCP, который устанавливает этот параметр при подключении компьютера к сети. При установке фиксированного адреса DNS-сервера (и адреса запасного DNS-сервера), для разрешения имен будет использоваться, именно тот DNS-сервер, адрес которого будет указан в соответствующем поле окна.

Более подробно процесс установки и настройки протокола TCP/IP изложен в [7,10,11].

2.11. Итоги главы

1. Стек протоколов TCP/IP является основным стандартом, используемый для взаимодействия распределенных в сети компонентов программных систем. TCP/IP поддерживается большинством современных операционных систем и применяется

практически во всех системных и прикладных программных системах, работающих в сети. Описание всех протоколов TCP/IP содержится в документах именуемых RFC и поддерживаемых группой IETF.

2. Модель TCP/IP является четырехуровневой и в основном согласуется с моделью ISO/OSI. На Уровне доступа к сети используются протоколы, обеспечивающие создание локальных сетей или соединений с глобальными сетями. Наиболее популярными протоколами этого уровня являются Ethernet и PPP. Основным протоколом Межсетевого уровня является IP, но используются и другие протоколы: ICMP (для транспортировки служебной информации и диагностики), ARP (для разрешения MAC-адресов) и т.д. Транспортный уровень обеспечивается протоколами UDP (ненадежный протокол, ориентированный на сообщения) и TCP (надежный протокол ориентированный на соединение). На Прикладной уровне находятся службы TCP/IP и прикладные системы пользователей.
3. Протоколу IP (или точнее IPv4) в семействе протоколов TCP/IP отведена центральная роль. Его основной задачей является доставка дейтаграмм. Протокол считается ненадежным и не поддерживающим соединения. Для доступа к хостам IP использует собственную систему адресации. IP-адресом является последовательность из 32 битов, состоящая из адреса сети и адреса хоста в данной сети. Количество бит отведенных на адрес сети и на адрес хоста зависит от используемой модели адресации.
4. Новая версия протокола IP называется IPv6. Главным отличием протокола IPv6 является применение 128-битных адресов, что позволяет сделать все IP-адреса уникальными в глобальном смысле.
5. Протоколы Транспортного уровня являются пограничными протоколами между прикладной программной системой и стеком протоколов TCP/IP. Любой прикладной процесс протоколы Транспортного уровня идентифицируют с помощью номера порта. Все номера портов разбиты на три группы: хорошо известные номера портов (отводятся для базовых служб TCP/IP), зарегистрированные номера портов (используются известными промышленными системами) и динамически распределяемые или эфемерные порты (выделяются операционной системой динамически, по мере необходимости).
6. Для идентификации прикладного процесса в сети используется понятие сокета. Сокет – это совокупность IP-адреса и номера порта. Кроме того, различают сокеты UDP и сокеты TCP.
7. Большинство реализаций TCP/IP поддерживают интерфейс внутренней петли, позволяющий процессам, находящимся на одном хосте, обмениваться данными по протоколу TCP/IP. При этом дейтаграммы не выходят за пределы TCP/IP-интерфейса хоста. Внутренняя петля применяется, в основном, для отладки распределенных приложений.

С ее помощью на одном компьютере можно смоделировать работу процессов в сети.

8. Для доступа прикладных процессов к процедурам TCP/IP, операционные системы предоставляют специальные API. Наиболее распространенными программными интерфейсами являются API сокетов и интерфейс RPC. API сокетов описан в стандарте POSIX и поддерживается большинством операционных систем. Наиболее распространенной версией RPC является версия RPC Sun Microsystems. Развитием технологии RPC в Windows являются технологии COM и DCOM, а в Java-технологиях механизм RMI.
9. Программную реализацию протоколов Прикладного уровня TCP/IP называют службами. Службы реализуются в виде серверов, предоставляющих услуги другим процессам. Наиболее часто используемыми службами являются: DHCP, DNS, NBT, Telnet, FTP, WWW (протокол HTTP) и служба электронной почты (протоколы SMTP, POP3, IMAP4).
10. Для диагностики TCP/IP и управления сетью используются специальные программы, называемые сетевыми утилитами. Перечень утилит и их параметры зависят от конкретной реализации TCP/IP.
11. Стек протоколов TCP/IP, как правило, устанавливается на компьютер вместе с операционной системой. Работа системного администратора сводится к достаточно простой настройке TCP/IP.