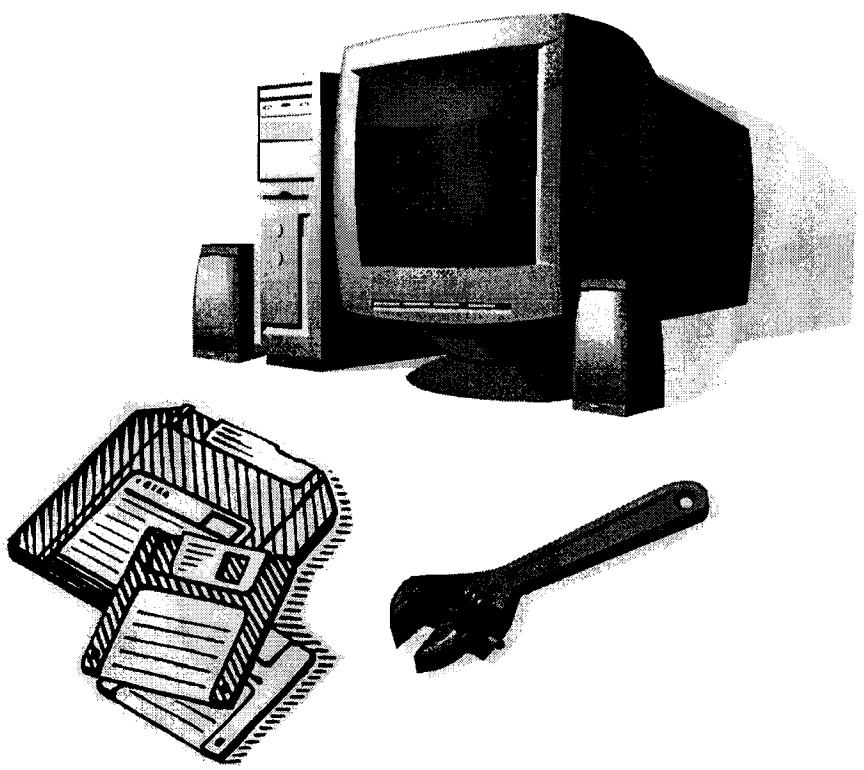
Е.П. Дятлова, А.И. Новиков

Структура и принцип работы вычислительных сетей АСУ

Учебное пособие



Санкт-Петербург
2009

Федеральное агентство по образованию

Государственное образовательное учреждение
высшего профессионального образования

«Санкт-Петербургский государственный технологический
университет растительных полимеров»

Е. П.Дятлова, А. И.Новиков

Структура и принцип работы вычислительных сетей АСУ

Учебное пособие

Санкт-Петербург
2009

ББК 32.965
Д 998
УДК 681.3
Дятлова Е.П., Новиков А.И. СТРУКТУРА И ПРИНЦИП РАБОТЫ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ АСУ: учеб. пособие/ ГОУВПО СПб ГТУ РП. - СПб.,2009:78 с.

В учебном пособии рассмотрены основные структуры и принципы работы вычислительных сетей АСУ.

Учебное пособие предназначено для использования студентами специальности 220200 при изучении дисциплин «Вычислительные сети АСУ», «Проектирование автоматизированных систем», «Автоматизация технологических процессов», а также при выполнении курсовых проектов и выпускных квалификационных работ.

Рецензенты:

профессор кафедры автоматизации процессов химических производств Санкт-Петербургского технического университета, д-р техн. наук Л.А.Русинов;

доцент кафедры автоматизации химико-технологических процессов Санкт-Петербургского технологического университета растительных полимеров, канд. техн. наук Ю.С.Жукова.

Рекомендовано к изданию Редакционно-издательским советом Санкт-Петербургского государственного технологического университета растительных полимеров в качестве учебного пособия.

1. СЕТИ ETHERNET

1.1. Классификация сетей Ethernet

Наибольшее распространение в домашних и промышленных сетях на сегодняшний день получила технология Ethernet. Эта технология была разработана в 1970 г, а в 1980 г. на ее основе был создан стандарт IEEE 802.3. Данная технология имеет три реализации: непосредственно Ethernet (рис. 1.1), Fast Ethernet (рис. 1.2) и Gigabit Ethernet – со скоростями 10, 100 и 1000 Мбит в секунду соответственно. Каждая последующая реализация является модернизацией предыдущей и работает на больших скоростях.

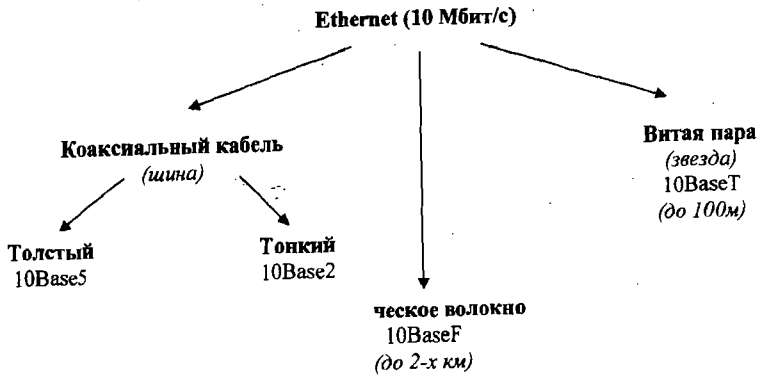


Рис. 1.1. Структура Ethernet 10Мбит/с

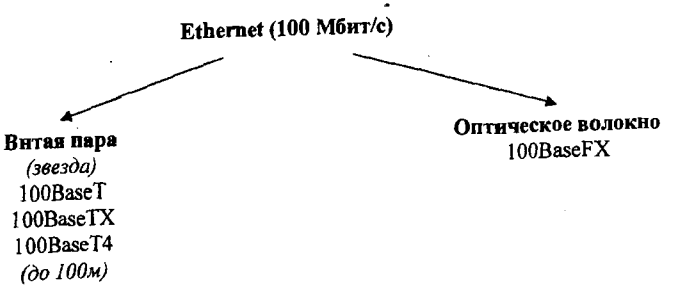


Рис. 1.2. Структура Ethernet 100Мбит/с

Gigabit Ethernet представлен стандартом 1000BaseT со скоростью передачи данных до 1000Мбит/с (1Гигабит/с) и длинной одного сегмента до 100 м. Для построения сетей на основе Gigabit Ethernet должна быть использована витая пара (см. раздел 1.4) не ниже 5-й категории.

1.2. Обозначение стандартов сети Ethernet

XBaseY,

где **X** - пропускная способность сети (Мбит/с);

Y - максимальная длинна сегмента (в сотнях метров)

или **T**- twisted pair (витая пара)

или **F**-fiber optic (оптическое волокно).

1.3. Коаксиальный кабель

Коаксиальный кабель (рис. 1.3) делится на толстый и тонкий в зависимости от диаметра [1]. На концах каждого из отрезков кабеля монтируются разъемы BNC.

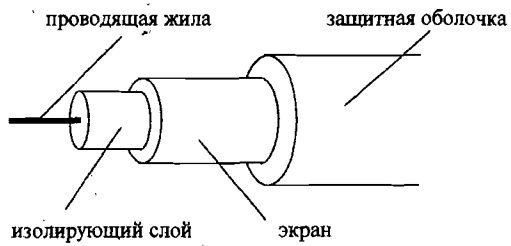


Рис. 1.3. Коаксиальный кабель

Сеть на базе коаксиального кабеля строится по типологии ШИНА (см. разд. 1.7). Для соединения отрезков коаксиального кабелей используется Т-коннекторы (рис. 1.4).

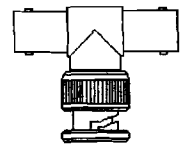


Рис. 1.4. Т-коннектор

На свободные концы коаксиального кабеля **ОБЯЗАТЕЛЬНО** устанавливаются терминаторы, представляющие собой гасители сигнала сопротивлением 50 Ом.

1.4. Витая пара

Витая пара представляет собой два провода, переплетенных вместе, что позволяет снизить уровень помех при передаче данных. Кабель Ethernet, который называется кабель «витая пара», представляет собой четыре (реже две) витые пары в одной оплетке. Витая пара может быть экранированной (обозначается STP) и не экранированной (обозначается UTP). Для обжима витой пары используются коннекторы RJ-45 (рис. 1.5).

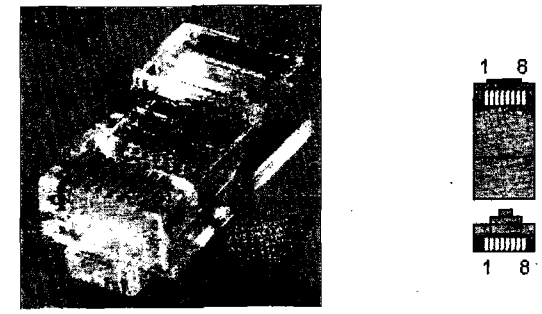
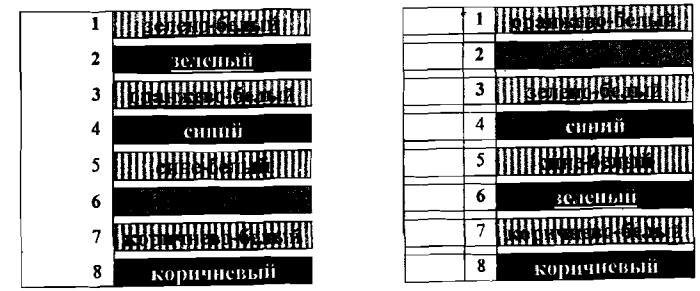


Рис. 1.5. Коннектор RJ-45

1.5. Правила обжима витой пары

Существует два способа установки коннектора RJ-45 на кабеле «витая пара» [1, 2]: EIA/TIA-568A (рис 1.6а) и EIA/TIA-568B (рис 1.6б).



а) EIA/TIA-568A б) EIA/TIA-568B

Рис. 1.6. Варианты обжима витой пары

Кроме этого, правила обжима кабеля зависят от того, какие устройства необходимо соединить:

1) Компьютер – Hub

В этом случае на противоположных концах кабеля используются одинаковые способы обжима (рис. 1.7), либо EIA/TIA-568A и EIA/TIA-568A, либо EIA/TIA-568B и EIA/TIA-568B.

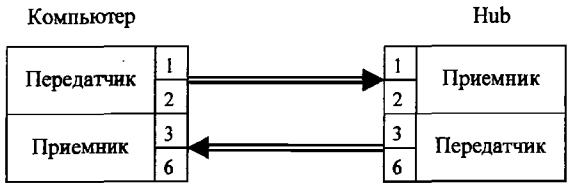


Рис. 1.7. Соединение Компьютер – Hub

2) Компьютер – Компьютер

В этом случае на противоположных концах кабеля используются различные способы обжима (рис. 1.8), т.е. и EIA/TIA-568A и EIA/TIA-568B, а кабель называется перекрестным (кроссовым).

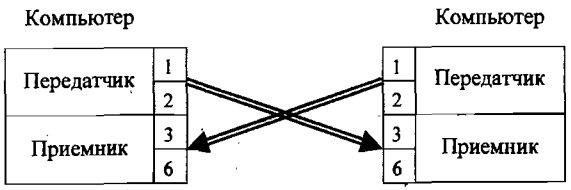


Рис. 1.8. Соединение Компьютер – Компьютер

1.6. Оптическое волокно

Оптическое волокно является более дорогим по сравнению с витой парой и коаксиальным кабелем, а также требует определенной квалификации и инструмента для его монтажа, но зато позволяет прокладывать участки кабеля (без дополнительных усилителей) длиной до 250 км. Проходя по оптическому кабелю, свет многократно отражается (рис. 1.9), при этом

потери сигнала малы. Как правило, используется в оптических магистралях для передачи больших объемов данных на дальние расстояния. Поэтому в большинстве случаев в оптоволоконной сети имеет место соединение «точка-точка», когда по оптоволокну соединяются лишь два устройства, а дальше данные расходятся к своим адресатам по витой паре (рис. 1.10). Подобная топология, также как и для сети на основании чисто витой пары, называется «звезда».

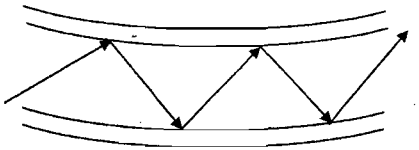


Рис. 1.9. Оптическое волокно

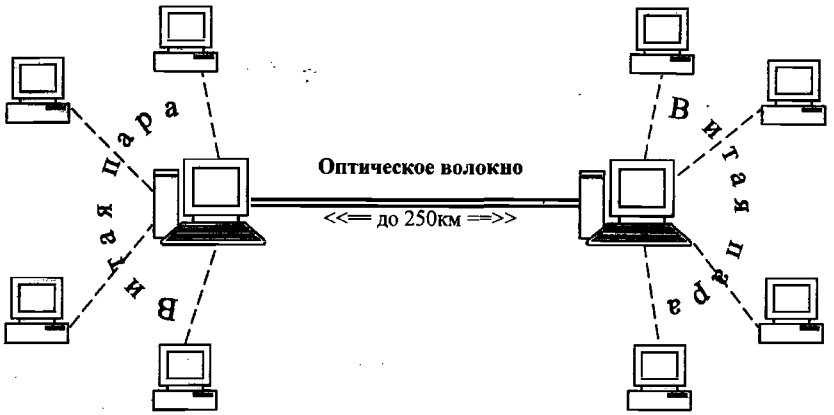


Рис. 1.10. Оптоволоконная магистраль

1.7. Топологии сетей

Звезда

В настоящее время «звезда» (рис. 1.11) является наиболее часто встречающейся топологией для массового непромышленного использования. Связано это в основном с распространением Ethernet.

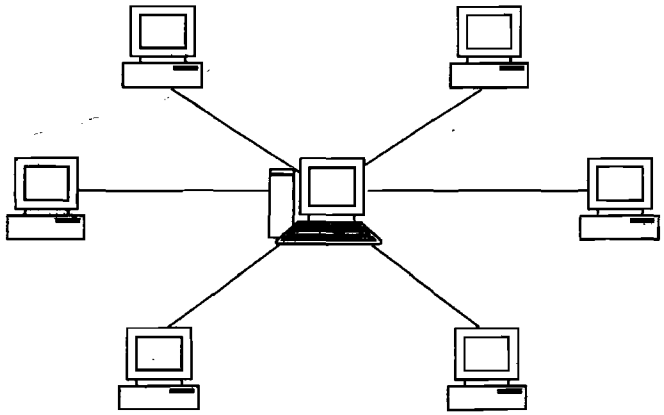


Рис. 1.11. Топология «звезда»

Компьютеры соединяются через центральное устройство – сервер. При выходе из строя одного из компьютеров, сеть продолжает работать, но при выходе из строя сервера связь между компьютерами пропадает.

Шина

Топология «шина» (рис. 1.12) используется для построения сетей Ethernet 10Base5 и 10Base2, которые в настоящее время считаются устаревшими. Подсоединение компьютеров осуществляется в разрыв кабеля с помощью Т-коннектора. При отсоединении компьютера Т-коннектор удаляется, а на его место устанавливается перемычка, соединяющая два сегмента разорванного кабеля. На концах шины **ОБЯЗАТЕЛЬНО** устанавливаются терминаторы.

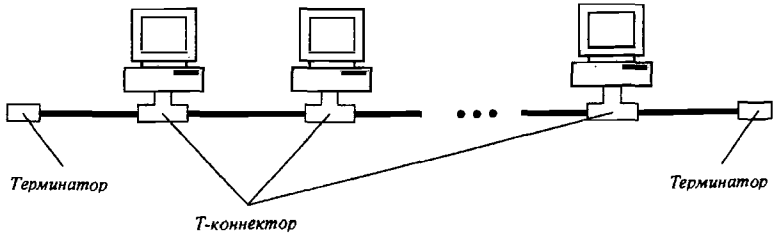


Рис. 1.12. Топология «шина»

Кольцо

«Кольцо» используется в сетях отличных от Ethernet, например Token Ring, где по сети постоянно циркулирует короткий блок данных (маркер), получаемый каждой рабочей станцией по очереди.

Можно выделить два подтипа данной топологии: одностороннее кольцо (рис. 1.13, рис. 1.14).

В одностороннем кольце при выходе из строя одного из сегментов кабеля, соединяющего рабочие станции, выходит из строя вся сеть. Эта проблема решается передачей сообщений в обоих направлениях (рис. 1.14) и при выходе из строя одного из сегментов кабеля, соединяющего рабочие станции, сообщения могут передаваться в обратном направлении по второму кольцу. Двухстороннее кольцо является более надежным т.к. даже при выходе из строя одновременно нескольких элементов сети отдельные ее части продолжают функционировать. Но двойное кольцо является более дорогостоящим по сравнению с односторонним из-за большего использования кабеля.

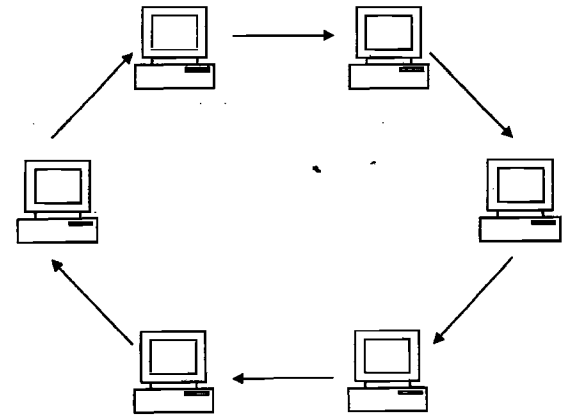


Рис. 1.13. Одностороннее кольцо

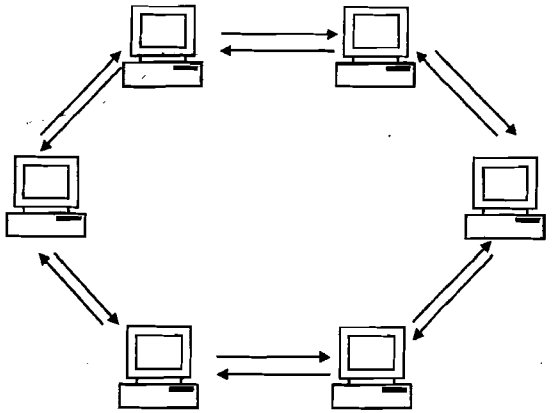


Рис. 1.14. Двухстороннее кольцо

Смешанная топология

Сеть может быть смешанной, т.е. содержать в себе признаки различных топологий (рис. 1.15). В некоторых литературных источниках подобную структуру сети выделяют в отдельную (четвертую) топологию.

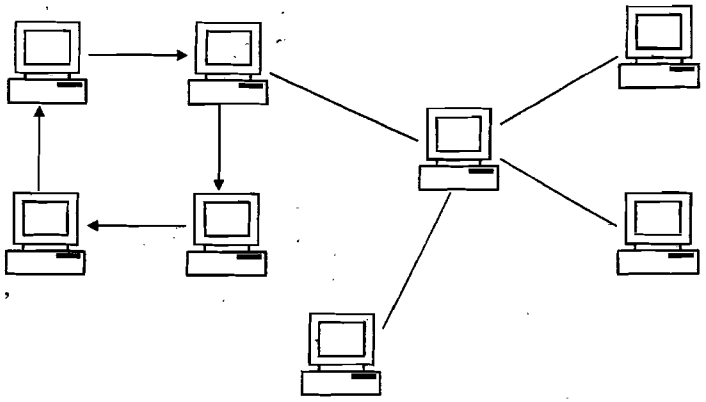


Рис. 1.15. Пример смешанной топологии

При этом узел, принадлежащий одновременно двум или более сетям, является шлюзом либо мостом.

Мост (bridge) - устройство, соединяющее две отдельные сети или два сегмента одной сети, использующих одинаковые протоколы.

Шлюз (gateway) - более сложное устройство, объединяющее разнородные сети.

1.8. Концентратор Hub

Hub – концентратор является центральным звеном топологии «звезда» (рис. 1.16) [1]. Имеет восемь разъемов RJ45 или большее их число, кратное восьми. Особенности работы hub является то, что, получив сообщение от одного компьютера, он передает его на все компьютеры. Таким образом, при передаче сообщения одним из компьютеров другие передавать не могут. Hub работает в режиме полудуплекса.

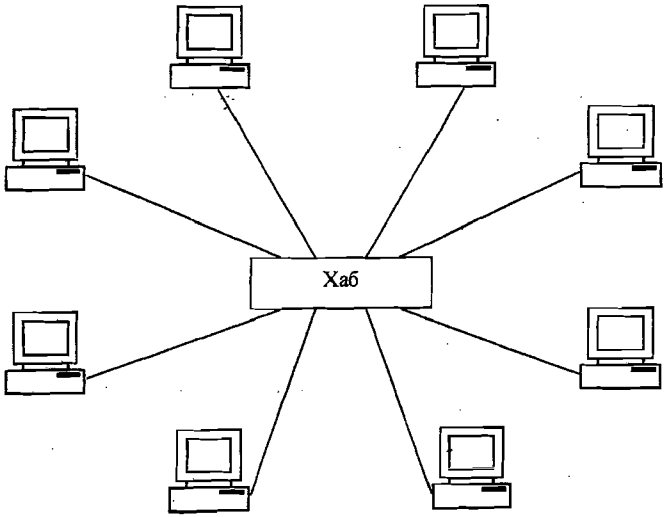


Рис. 1.16. Центральное звено топологии «звезда»

1.9. Коммутатор Switch

Switch (коммутатор) - аналогичен hub, за исключением того, что сообщение посылается на конкретный компьютер. Switch может работать в режиме полного дуплекса.

1.10. Репитеры

Репитеры (повторители) - используются для увеличения длины сети (выступают в роли усилителя сигнала).

1.11. Настройка сети в Windows XP

Настройка имени компьютера и рабочей группы

Имя компьютера и рабочей группы задается в свойствах системы. Для вызова окна свойств системы необходимо щелкнуть правой клавишей мыши по значку «**мой компьютер**» на рабочем столе и в появившемся контекстном меню выбрать пункт **свойства** (рис. 1.17).

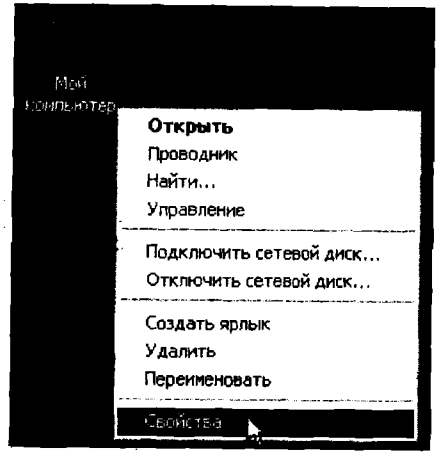


Рис. 1.17. Контекстное меню «Мой компьютер»

В появившемся окне свойств системы (рис. 1.18) необходимо выбрать закладку «**Имя компьютера**» и далее нажать кнопку «**Изменить**».

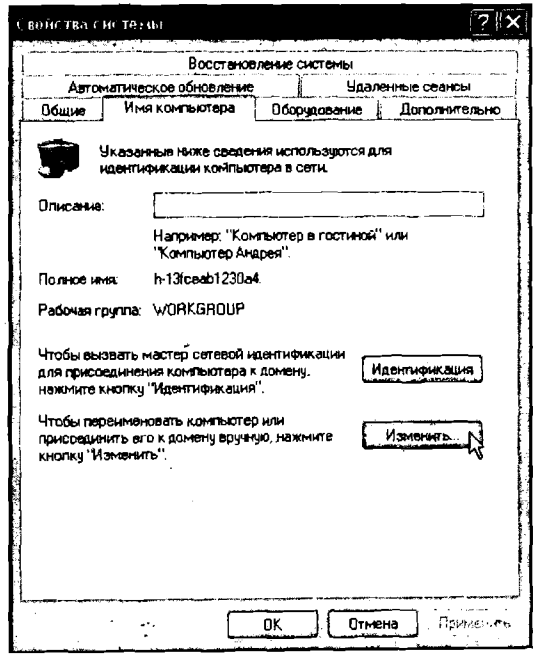


Рис. 1.18. Свойства системы

Имена компьютеров в сети должны быть уникальными, а рабочая группа одинаковой.

Настройка IP-адреса и маски подсети

Для задания IP-адреса и маски подсети необходимо:

а) Щелкнуть правой клавишей мыши по значку **сетевое окружение** и выбрать пункт меню свойства (рис. 1.19).

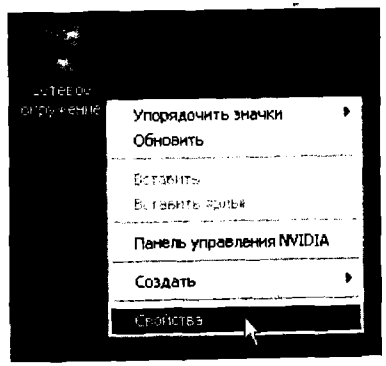


Рис. 1.19. Контекстное меню сетевого окружения

В открывшемся окне «Сетевые подключения» (рис. 1.20) будет отображен список всех сетевых устройств, установленных на компьютере.

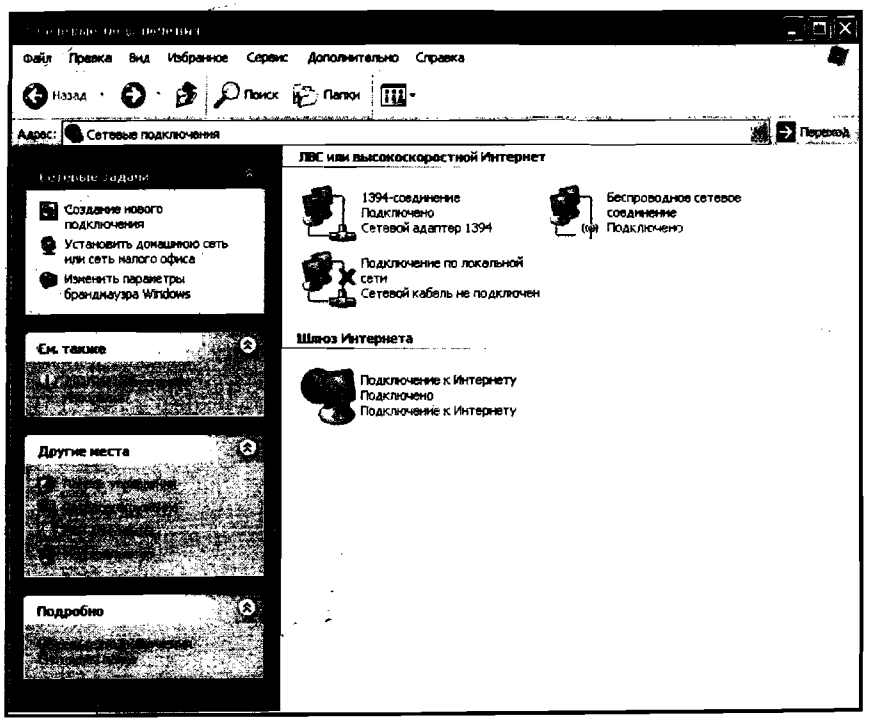


Рис. 1.20. Сетевые подключения

Текущее состояние каждого из подключений можно определить по внешнему виду его ярлыка, а также по надписи под названием **подключения**.

Подключения могут иметь три основных состояния:

- Подключено *(цветной значок в виде двух компьютеров).*

- Отключено *(черно-белый значок в виде двух компьютеров).*

- Сетевой кабель не подключен (Сеть не найдена)

(цветной значок в виде двух компьютеров с красным крестом).

Переход из состояния отключено в состояние подключено/сеть не найдена осуществляется выбором пункта подключить контекстного меню для соответствующего соединения.

Вызов контекстного меню осуществляется щелчком правой клавиши мыши по значку соответствующего соединения.

Переход из состояния подключено/сеть не найдена в состояние отключено осуществляется выбором пункта меню отключить.

Переход между состояниями подключено и сеть не найдена осуществляется автоматически в зависимости от того установлено ли физическое соединение.

б) Выбрать из контекстного меню требуемого соединения пункт свойства.

В появившемся окне (рис 1.21) установить галочку «Вывести значок в области уведомления» («Отображать состояние подключения»).

Из списка протоколов выбрать протокол TCP/IP, дважды щелкнув по нему мышкой.

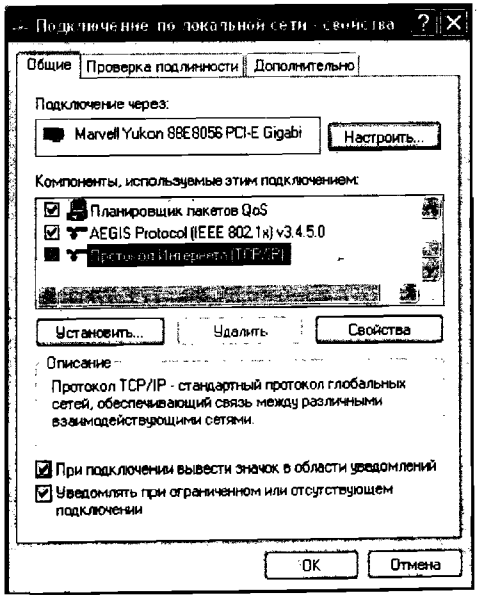


Рис. 1.21. Свойства подключения

в) В появившемся окне (рис. 1.22) задать IP-адрес и маску подсети вручную.

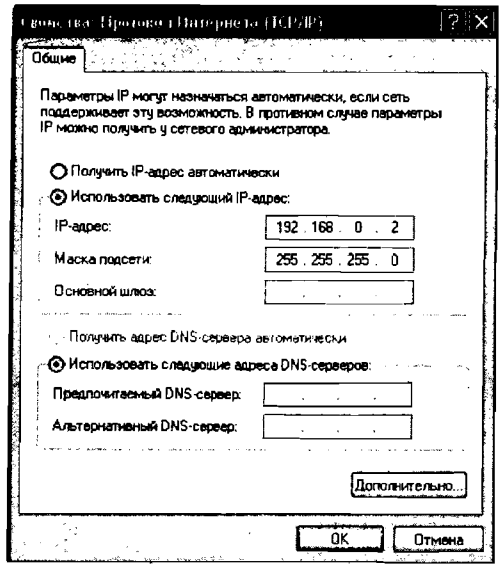


Рис. 1.22. Свойства протокола TCP/IP

IP-адреса компьютеров в сети должны быть уникальным, а маска подсети одинаковой.

Например:

Имя компьютера	Comp1	Comp2	Comp3	...
Рабочая группа	WORKGROUP			
IP	192.168.0.1	192.168.0.2	192.168.0.3	...
Маска подсети	255.255.255.0			

г) Для подключения к сети **Internet**, необходимо также указать **шлюз** и **DNS-сервер** (рис. 1.23). Как правило, в малых сетях адреса **шлюза** и **DNS-сервера** совпадают, что свидетельствует о совмещении обеих функций в одном устройстве (сервере). По умолчанию это адрес **192.168.0.1**. В этом случае целесообразно переименовать **Comp1** в **Server**.

За более подробной информацией обращайтесь к вашему провайдеру.

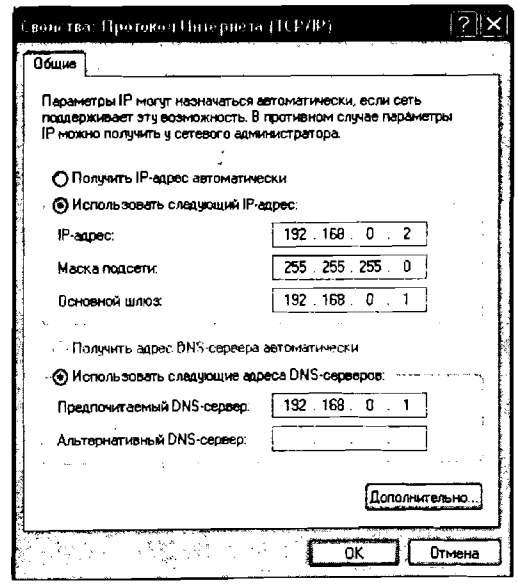


Рис. 1.23. Указание шлюза и сервера DNS

1.12. Режимы передачи данных

Существует три режима передачи данных: односторонний (симплексный), дуплексный и полудуплексный.

При односторонней передаче данных (рис. 1.24) одно устройство всегда является только передающим, а другое только принимающим.

Примером симплексного режима может служить телевидение.

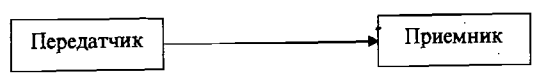


Рис. 1.24. Односторонний (симплексный) режим передачи данных

При дуплексной передаче (рис. 1.25) данные передаются в обоих направлениях одновременно.

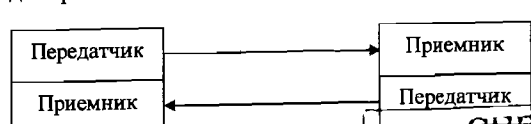


Рис. 1.25. Дуплексный режим передачи данных

При полудуплексной передаче (рис. 1.26) данные передаются в обоих направлениях, но по очереди, по одной паре проводов.

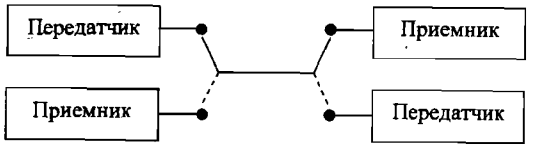
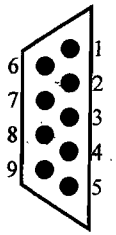


Рис. 1.26. Полудуплексный режим передачи данных

2. ИНТЕРФЕЙСЫ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

2.1. Последовательный интерфейс RS-232

RS-232 представляет из себя обычный **COM**-порт персонального компьютера (рис. 2.1).



Контакты разъема:

2 – RxD - передача данных;

3 – TxD - прием данных;

5 – GROUND - земля.

Рис. 2.1. Внешний вид **COM**-разъема *(со стороны компьютера)*

RS-232 позволяет соединить между собой два компьютера (или два любые другие устройства) на расстояние до 3-5м.

Передача данных (рис. 2.2) может осуществляться в обоих направлениях одновременно (полный дуплекс).

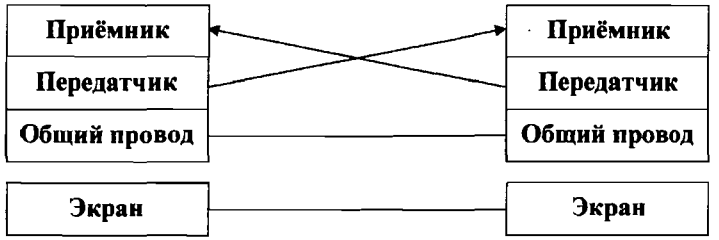


Рис. 2.2. Соединение по RS-232

В промышленности большее распространение получил последовательный интерфейс RS-485, позволяющий соединить до 16 устройств на расстояние в несколько сотен метров. Соединение осуществляется по витой паре по двум (полудуплекс) или четырем (полный дуплекс) проводам.

2.2. Hyper Terminal

Проверить соединение компьютеров через COM-порты можно программой Hyper Terminal, расположенной в меню ПУСК/Программы/Стандартные/Связь.

После запуска программы (рис. 2.3) необходимо:

указать имя соединения (желательно без использования русских букв и пробелов);

выбрать порт, к которому физически подсоединен кабель (например COM1);

указать параметры порта (скорость, биты данных, четность, стоповые биты), одинаковые для обоих компьютеров.

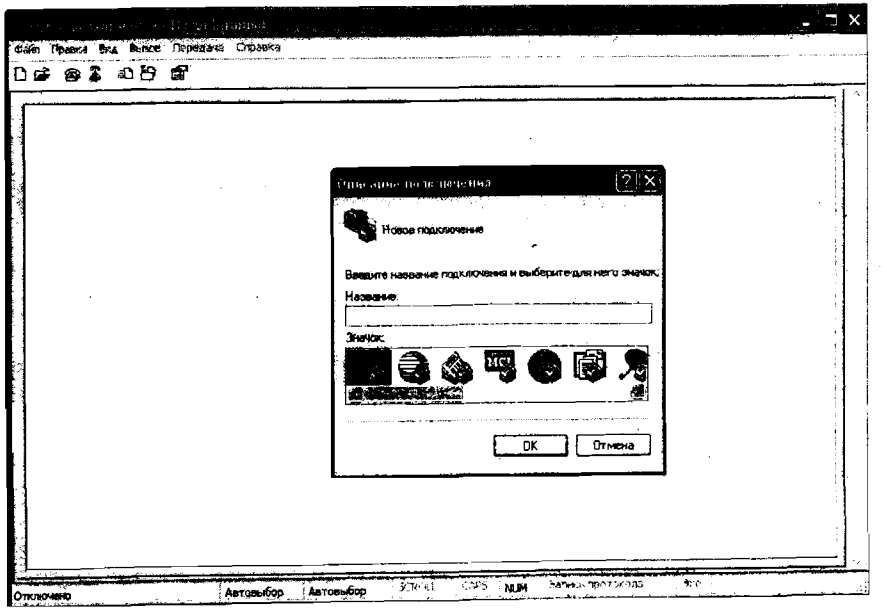
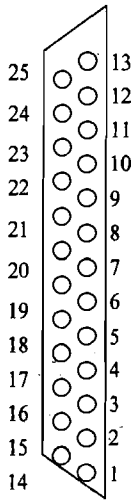


Рис. 2.3. Внешний вид окна программы Hyper Terminal

В случае если соединение установлено правильно, текст, вводимый в окне одного компьютера, будет виден в окне другого компьютера. Также программа позволяет передавать через COM-порт файлы.

2.3. Параллельный порт LPT (порт принтера)

Параллельный LPT-порт (рис. 2.4). позволяет передавать одновременно 8 бит данных и принимать 5 бит состояния.



Контакты разъема:

2-9 – D0-D7 –выходные биты данных;

18 – питание +5В;

10-13,15 – входные биты состояния;

18-25 – GND – корпус.

Рис. 2.4. Внешний вид **LPT**-разъема *(со стороны компьютера)*

Не предназначен для соединения компьютер-компьютер, хотя такая связь и возможна (рис. 2.5). В этом случае в каждом из направлений одновременно можно передавать не более пяти бит данных.

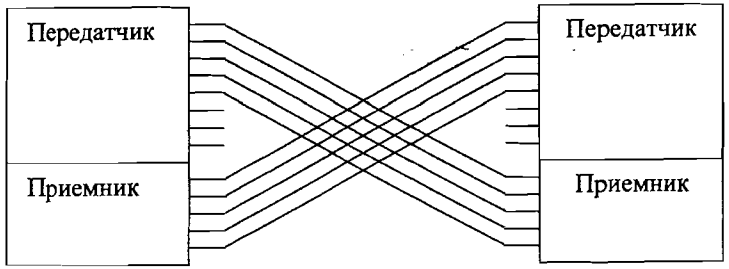


Рис. 2.5. Соединение двух компьютеров по LPT

3. INTERNET

В данной главе рассматривается сеть Internet, как наиболее распространенная (глобальная) сеть. Описанные способы маршрутизации приведены в общем виде и применительны также для других случаев. Internet является лишь объединением нескольких типов сетей, таких как рассмотренный выше Ethernet, и нескольких десятков протоколов передачи данных.

Возможности INTERNET

Существует 7 основных путей использования INTERNET:

Электронная почта. С помощью почтовых программ Outlook Express, Netscape Messenger и др.

1. Отправка и получение файлов с помощью FTP (File Transfer Protocol)
2. Чтение и посылка текстов в USENET
3. Поиск информации через GOPHER и WWW (World Wide Web)
4. Удаленное управление - запрос и запуск программ на удаленном компьютере.
5. Chat-разговор с помощью сети IRC и Электронной почты
6. Игры через INTERNET

3.1. Доменные имена

Когда вы обращаетесь на Web или посылаете e-mail, вы используете доменное имя [3]. Например, адрес **http://www.microsoft.com/** содержит доменное имя microsoft.com. Аналогично e-mail-адрес **myname@mail.ru** содержит доменное имя mail.ru.

В доменной системе имен реализуется принцип назначения имен с определением ответственности за их подмножество соответствующих сетевых групп. И если каждая группа придерживается этого простого правила и всегда получает подтверждение, что имена, которые она присваивает, единственны среди множества ее непосредственных

подчиненных, то никакие две системы, где бы те ни находились в сети Интернет, не смогут получить одинаковые имена.

Также уникальны адреса, указываемые на конвертах при доставке писем обычной почтой. Таким образом, адрес на основе географических и административных названий однозначно определяет точку назначения.

Домены тоже имеют аналогичную иерархию. В именах домены отделяются друг от друга точками: companya.msk.ru, companyb.spb.ru. В имени может быть различное количество доменов, но обычно их не больше пяти. По мере движения по доменам в имени слева направо, количество имен, входящих в соответствующую группу, возрастает.

Каждый раз, когда вы используете доменное имя, вы также используете DNS-серверы для того, чтобы перевести буквенное доменное имя в IP-адрес на машинном языке.

В качестве примера рассмотрим адрес **www.pc.dpt1.company.spb.ru**.

Первым в имени стоит название рабочей машины — реального компьютера с IP-адресом. Это имя создано и поддерживается группой dpt1. Группа входит в более крупное подразделение company, далее следует домен spb — он определяет имена петербургской части сети, а ru — российской.

Каждая страна имеет свой домен. Так au — соответствует Австралии, be — Бельгии и т.д. Это географические домены верхнего уровня. Помимо географического признака используется тематический, в соответствии с которым существуют следующие доменные имена первого уровня:

- com — обозначает коммерческие предприятия;
- edu — образовательные;
- gov — государственные;
- mil — военные;
- net — сетевые;
- org — учреждения других организаций и сетевых ресурсов.

Внутри каждого доменного имени первого уровня находится целый ряд доменных имен второго уровня. Домен верхнего уровня располагается в имени правее, а домен нижнего уровня — левее.

Например, на рис 3.1 показана структура адреса ряда организаций на примере российского домена.

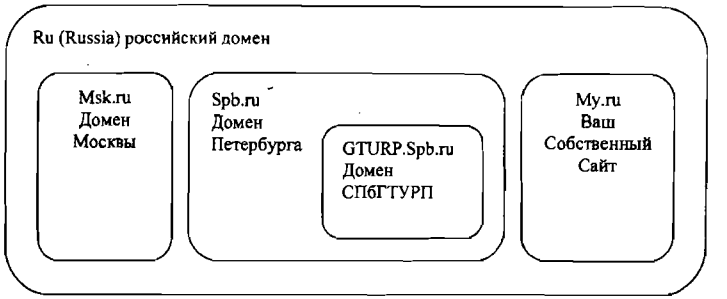


Рис. 3.1. Структура адреса ряда организаций на примере российского домена.

Рассмотрим адрес **http://www.gturp.spb.ru/**. Домен верхнего уровня ru указывает на то, что адрес принадлежит российской части Интернета, spb — определяет город, следующий уровень — домен конкретной организации. В принципе, в имени может быть любое число доменов.

3.2. DNS-сервер

DNS-сервер принимает запрос на конвертацию доменного имени в IP-адрес. При этом DNS-сервер выполняет следующие действия:

- отвечает на запрос, выдав IP-адрес, поскольку уже знает IP-адрес запрашиваемого домена;
- контактирует с другим DNS-сервером для того, чтобы найти IP-адрес запрошенного имени. Этот запрос может проходить по цепочке несколько раз;
- выдает сообщение: «Я не знаю IP address домена, запрашиваемого вами, но вот IP address DNS-сервера, который знает больше меня»;
- сообщает, что такой домен не существует.

Представим, что вы набрали адрес **http://www.pc.dpt1.company.com/** в вашем браузере, который имеет адрес в домене верхнего уровня COM (рис. 3.2). В простейшем варианте ваш браузер контактирует с DNS-сервером для того, чтобы получить IP-адрес искомого компьютера, и DNS-сервер возвращает искомый IP-адрес (рис. 3.3).

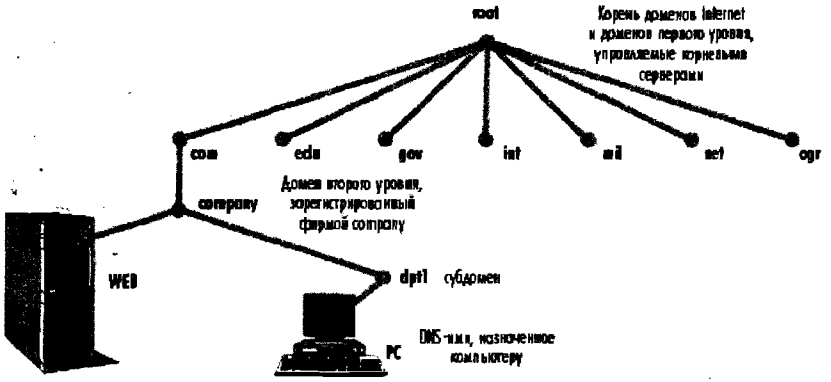


Рис. 3.2. Структура адреса для pc.dpt1.company.com

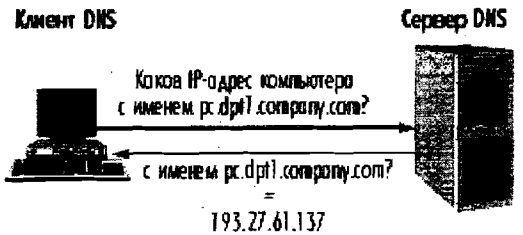


Рис. 3.3. Запрос к DNS-серверу на получение IP-адреса

На практике в Сети, где объединены миллионы компьютеров, найти DNS-сервер, который знает нужную вам информацию, — это целая проблема. Иными словами, если вы ищете какой-то компьютер в Сети, то прежде всего вам необходимо найти DNS-сервер, на котором хранится

нужная вам информация. При этом в поиске информации может быть задействована целая цепочка серверов (рис. 3.4).

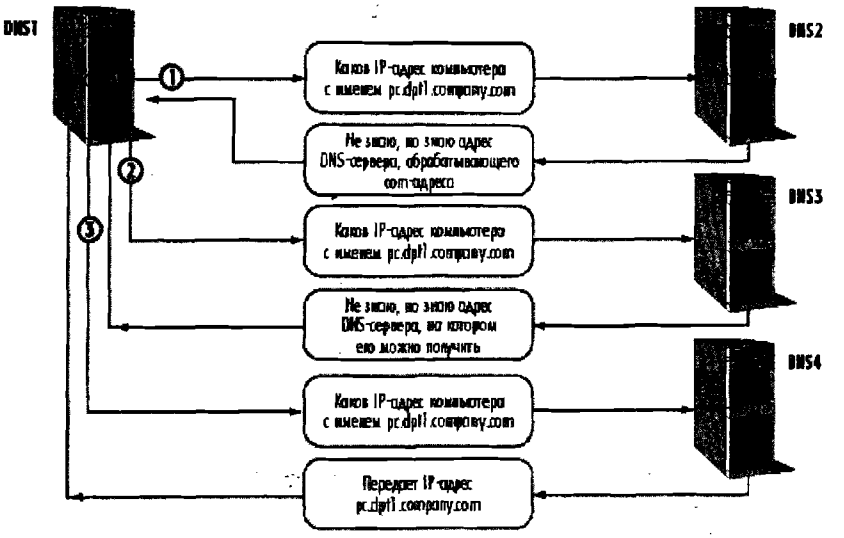


Рис. 3.4. Поиск DNS-сервера

Предположим, что тот DNS-сервер, к которому вы обратились (на рис. 3.4 он обозначен как DNS1), не имеет нужной информации. DNS1 начнет поиск IP-адреса с обращения к одному из корневых DNS-серверов. Корневые DNS-серверы знают IP-адреса всех DNS-серверов, отвечающих за доменные имена верхнего уровня (COM, EDU, GOV, INT, MIL, NET, ORG и т.д.).

Например, ваш сервер DNS1 может запросить адрес у корневого DNS-сервера. Если корневой сервер не знает данного адреса, возможно, он даст ответ: «Я не знаю IP-адреса для **http://www.pc.dpt1.company.com/**, но могу предоставить IP-адрес COM DNS-сервера».

После этого ваш DNS посылает запрос на COM DNS с просьбой сообщить искомый IP-адрес. Так происходит до тех пор, пока не найдется DNS-сервер, который выдаст нужную информацию.

Одна из причин, по которой система работает надежно, — это ее избыточность. Существует множество DNS-серверов на каждом уровне, и поэтому, если один из них не может дать ответ, наверняка существует другой, на котором есть необходимая вам информация. Другая технология, которая делает поиск более быстрым, — это система кэширования. Как только DNS-сервер выполняет запрос, он кэширует полученный IP-адрес. Однажды сделав запрос на корневой DNS (root DNS) и получив адрес DNS-сервера, обслуживающего COM-домены, в следующий раз он уже не должен будет повторно обращаться с подобным запросом. Кэширование происходит с каждым запросом, что постепенно оптимизирует скорость работы системы. Несмотря на то, что пользователям работа DNS-сервера не видна, эти серверы каждый день выполняют миллиарды запросов, обеспечивая работу миллионов пользователей.

Рассмотрим сеть небольшой компании (рис. 3.5), подключенной к Internet. Пусть пользователь станции PC1 хочет загрузить страничку сайта, расположенного по адресу www.gturp.spb.ru. Для этого необходимо получить IP-адрес сервера, на котором хранится страничка. PC1 отправляет запрос (1) на получение IP-адреса DNS-серверу. DNS-сервер возвращает ответ (2) с результатом обработки запроса (в случае необходимости DNS-сервер может перенаправить запрос (1') на другие DNS-сервера, получая (2') от них требуемую информацию). Если в результате обработки запроса DNS-серверу удалось найти IP-адрес соответствующего сервера, то PC1 отправляет запрос (3) на данный сервер и получает в ответ (4) запрашиваемую информацию.

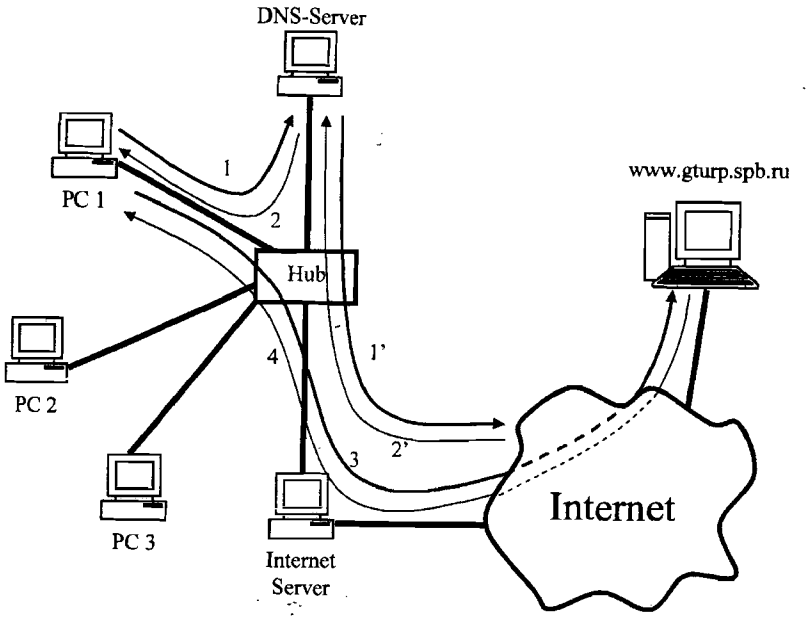


Рис. 3.5. Принцип работы DNS-сервера:

1 – запрос на получение IP-адреса по имени www.gturp.spb.ru;
2 – передача IP-адреса www.gturp.spb.ru;
3 – запроса на получение данных интернет странички ;
4 – интернет страничка СПбГТУРП;
1' – перенаправление запроса на получение IP-адреса по имени;
2' – ответ на перенаправленный запрос (1')

Нужно заметить, что запрос DNS может быть и обратным, т.е. получать доменное имя по IP-адресу.

3.3. Маршрутизация в сетях IP

Маршрутизатор (Router) - работает на сетевом уровне, определяя путь (маршрут) для пересылки проходящих через него пакетов. Может выполнять фильтрацию пакетов (решение о возможности передачи пакета принимается на основе анализа его содержимого). Маршрутизатором может служить специализированное оборудование или обычный компьютер. Каждый из интерфейсов маршрутизатора имеет собственный IP адрес.

Пример таблицы маршрутизации в ОС Windows:

Сетевой адрес	Маска сети	Адрес шлюза	Интерфейс	Метрика
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.10	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.10	192.168.0.10	1
192.168.0.1	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.0.255	255.255.255.255	192.168.0.10	192.168.0.10	1
224.0.0.0	224.0.0.0	192.168.0.10	192.168.0.1	1
255.255.255.255	255.255.255.255	192.168.0.10	192.168.0.10	1

Таблицы маршрутизации содержат записи, состоящие из следующих полей:

- адрес или множество адресов назначения (адрес и маска сети);
- адрес шлюза, которому передается пакет, адресованный в указанное множество;
- интерфейс маршрутизатора, который связывает его со шлюзом;
- метрика маршрута - число, определяющее его качество (расстояние до адресата, пропускная способность, надежность).

Множества адресов назначения в таблице маршрутизации могут содержаться одно в другом, (в частности множество, определенное первой строкой таблицы в примере ниже, включает все остальные возможные множества и определяет маршрут по умолчанию).

Если адрес приемника в IP пакете входит в несколько множеств, для определения маршрута пересылки используется множество минимальной мощности (наименьшее).

Для просмотра и настройки таблицы можно использовать команду ROUTE (ROUTE PRINT, ROUTE ADD, ROUTE DELETE...).

3.4. Типы адресов (MAC, IP, DNS)

Каждый компьютер в сети TCP/IP имеет адреса **трех** уровней:

- Локальный адрес узла, определяемый технологией, с помощью которой построена отдельная сеть, в которую входит данный узел. Для узлов, входящих в локальные сети - это **MAC-адрес** сетевого адаптера или порта маршрутизатора, например, 11-A0-17-3D-BC-01. Эти адреса

назначаются производителями оборудования и являются уникальными адресами, так как управляются централизовано. Для всех существующих технологий локальных сетей MAC-адрес имеет формат 6 байтов: старшие 3 байта - идентификатор фирмы производителя, а младшие 3 байта назначаются уникальным образом самим производителем.

- **IP-адрес**, состоящий из 4 байт, например, 109.26.17.100. Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации специального подразделения Internet (Network Information Center, NIC), если сеть должна работать как составная часть Internet. Обычно провайдеры услуг Internet получают диапазоны адресов у подразделений NIC, а затем распределяют их между своими абонентами.

Номер узла в протоколе IP назначается независимо от локального адреса узла. Деление IP-адреса на поле номера сети и номера узла - гибкое, и граница между этими полями может устанавливаться весьма произвольно. Узел может входить в несколько IP-сетей. В этом случае узел должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

- Символьный идентификатор-имя, например, SERV1.IBM.COM. Этот адрес назначается администратором и состоит из нескольких частей, например, имени машины, имени организации, имени домена. Такой адрес, называемый также **DNS**-именем, используется на прикладном уровне, например, в протоколах FTP или telnet.

3.5. Протоколы ARP и RARP

В протоколе IP-адрес узла, то есть адрес компьютера или порта маршрутизатора, назначается произвольно администратором сети и прямо не связан с его локальным адресом. Подход, используемый в IP, удобно использовать в крупных сетях и по причине его независимости от формата локального адреса, и по причине стабильности, так как в противном случае, при смене на компьютере сетевого адаптера это изменение должны бы были учитывать все адресаты всемирной сети Internet (в том случае, конечно, если сеть подключена к Internet'у).

Локальный адрес используется в протоколе IP только в пределах локальной сети при обмене данными между маршрутизатором и узлом этой сети. Маршрутизатор, получив пакет для узла одной из сетей, непосредственно подключенных к его портам, должен для передачи пакета сформировать кадр в соответствии с требованиями принятой в этой сети технологии и указать в нем локальный адрес узла, например его MAC-адрес. В пришедшем пакете этот адрес не указан, поэтому перед маршрутизатором встает задача поиска его по известному IP-адресу, который указан в пакете в качестве адреса назначения. С аналогичной задачей сталкивается и конечный узел, когда он хочет отправить пакет в удаленную сеть через маршрутизатор, подключенный к той же локальной сети, что и данный узел.

Для определения локального адреса по IP-адресу используется протокол разрешения адреса *Address Resolution Protocol, ARP*. Протокол ARP работает различным образом в зависимости от того, какой протокол канального уровня работает в данной сети - протокол локальной сети (Ethernet, Token Ring, FDDI) с возможностью широковещательного доступа одновременно ко всем узлам сети. Существует также протокол, решающий обратную задачу - нахождение IP-адреса по известному локальному адресу. Он называется реверсивный ARP - *RARP (Reverse Address Resolution Protocol)* и используется при старте бездисковых станций, не знающих в начальный момент своего IP-адреса, но знающих адрес своего сетевого адаптера.

В локальных сетях протокол ARP использует широковещательные кадры протокола канального уровня для поиска в сети узла с заданным IP-адресом.

Узел, которому нужно выполнить отображение IP-адреса на локальный адрес, формирует ARP запрос, вкладывает его в кадр протокола канального уровня, указывая в нем известный IP-адрес, и рассылает запрос широковещательно. Все узлы локальной сети получают ARP-запрос и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел формирует ARP-ответ, в котором указывает свой IP-адрес и свой локальный адрес и отправляет его уже направленно, так как в ARP-запросе отправитель указывает свой локальный адрес. ARP-запросы и ответы используют один и тот же формат пакета. Так как локальные адреса могут в различных типах сетей иметь различную длину, то формат пакета протокола ARP зависит от типа сети.

4. СЕТЕВЫЕ ПРОТОКОЛЫ

4.1. Протокол межсетевого взаимодействия IP

Основу транспортных средств стека протоколов TCP/IP составляет протокол межсетевого взаимодействия IP. К основным функциям протокола IP относятся присвоение и распознавание адресов, а также сборка и разборка пакетов. Последняя функция необходима в том случае, когда пакеты формируются в одной сети и передаются через другую сеть, в которой максимальная длина пакета меньше.

Структура пакета IP представлена на рис. 4.1. Назначение полей этого пакета следующее:

- *Номер версии* указывает версию протокола IP.
- *Длина заголовка* пакета IP меняется в зависимости от числа параметров.
- Поле *Тип сервиса* задает вид обслуживания. В нем предусмотрены биты указания задержки передачи, биты производительности и надежности локальной сети. Эти биты используются для задания приоритетов при формировании маршрута, например, в качестве критерия выбора маршрута может быть задана либо длина маршрута, либо надежность, либо сбалансированность трафика.
- Поле *Идентификатор пакета* используется для распознавания продублированных пакетов, а также при распознавании одинаковых пакетов, получившихся после фрагментации.
- *Идентификатор пакета* протокола верхнего уровня указывает, какому протоколу высокого уровня принадлежит пакет.
- *Флаг* указывает на целесообразность фрагментации пакета, а также наличие или отсутствие последующих пакетов при фрагментации.
- Поле *Смещение фрагмента* используется для указания количества байтов пакета, переданных до его фрагментации.

Рис. 4.1. Структура пакета протокола IP

- Поле *Время жизни* указывает продолжительность жизни данного пакета и задается протоколом IP источника передачи. На шлюзах и в других узлах сети по истечении каждой секунды от текущего времени жизни вычитается единица; единица вычитается также при каждой транзитной передаче (даже если не прошла секунда). При истечении срока жизни пакет аннулируется.
- *Контрольная сумма* рассчитывается по всему заголовку.
- *Адрес источника и адрес назначения* имеют одинаковую длину - 32 бита и одинаковую структуру.

IP - адрес состоит из 4 байтов (рис. 4.2) и записывается в виде четырех чисел, содержащихся в каждом байте, разделенных точками, например:

128.10.2.30 - десятичная форма представления адреса,

10000000 00001010 00000010 00011110 - двоичная форма.

Рис. 4.2. Структура пакета протокола IP

Адрес состоит из двух логических частей - номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая к номеру узла, определяется значениями первых бит адреса:

- Если адрес начинается с **0**, то сеть относят к классу **A**, номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Из этого следует, что сети класса А имеют

номера в диапазоне от 1 до 127. В сетях класса А количество узлов должно быть больше 2^{16} - не превышая 2^{24}.

- Если первые два бита адреса равны **10**, то сеть относится к классу **B** и является сетью средних размеров с числом узлов $2^8 - 2^{16}$. В сетях класса **B** под адрес сети отводится 14 бит, т. е. 2^{14} сетей, а под адрес узла-16.
- Если адрес начинается с последовательности **110**, то сеть класса **C**, в которой не может быть больше, чем 256 узлов, а под адрес сети отводится 21 двоичный разряд.
- Еще имеются два типа IP-адресов: адрес класса **D**, который начинается с последовательности **1110** и обозначает особый адрес **multicasting**, и адрес класса **E**, который начинается с кода **11110** и зарезервирован для будущих применений.

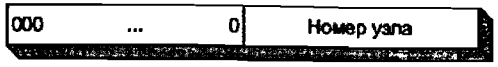
Такая структура адреса позволяет маршрутизаторам очень быстро извлекать из IP-адреса адрес сети.

До этого мы считали, что каждому узлу сети соответствует один IP-адрес. Однако, как быть с маршрутизатором, который связывает несколько сетей? Легко представить и такую ситуацию, когда компьютер входит в несколько сетей. В этом случае сетевой узел должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, **IP-адрес характернзует не отдельный компьютер или маршрутизатор, а одно сетевое соединение**.

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов:

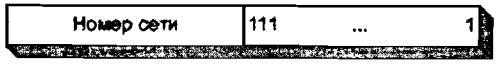
1) если IP-адрес состоит только из двоичных нулей, то он обозначает тот узел, который сгенерировал этот пакет:

2) если в поле номера сети стоят 0, то интерпретируется только номер узла и предполагается, что этот узел принадлежит той же самой сети, что и узел, который отправил пакет:



3) если все двоичные разряды IP-адреса равны 1. то этот пакет посылается всем узлам, но только относящимся к данной сети (ограниченное широковещательное сообщение - limited broadcast):

4) если сплошные 1(единицы) стоят только в поле адреса узла, то это означает широковещательное сообщение для узлов сети с заданным номером:



5) еще один адрес **127.0.0.1** зарезервирован для обозначения обратной связи – Loopback (замыкание на себя).

Уже упоминавшаяся форма IP-адреса - multicast - означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу, с номером, указанным в поле multicast. Узлы сами идентифицируют себя, то есть определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп. Такие сообщения в отличие от широковещательных называются мультивещательными.

Нужно заметить, что адреса компьютеров в подсетях классов A, B и C не могут быть нулевыми, номер сети узла так же не может быть нулевым.

В табл. 4.1 приведены диапазоны адресов для всех классов сетей.

Диапазоны адресов Таблица 4.1

Класс	Наименьший адрес	Наибольший адрес
A	1.0.0.0	126.0.0.0
B	128.0.0.0	191.255.0.0
C	192.0.0.0	223.255.255.0
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

Номер сети назначается специальным подразделением Network Information Center (NIC). Большие сети получают адреса класса A, средние - класса B, а маленькие - класса C. Получив номер, сетевой администратор часто сталкивается с проблемой, каким образом локализовать трафик в некоторых частях своей сети. Решение может быть найдено путем использования так называемых масок, которые позволяют делить сеть на подсети.

Маска - это число, двоичная запись которого содержит единицы в тех разрядах, которые должны интерпретироваться как адрес сети.

Например:

255.0.0.0 - маска для сети класса A,

255.255.0.0 - маска для сети класса B,

255.255.255.0 - маска для сети класса C.

Установив новое значение маски, можно заставить маршрутизатор интерпретировать адрес по-другому. Например, пусть сеть относится к классу B. В одной сети циркулирует единый трафик. Но среди всех станций сети есть некоторые, слабо взаимодействующие между собой. Эти станции желательно изолировать в разных сетях. Пусть это будут узел 129.34.17.15 и узел 129.34.20.01, которые в исходной ситуации относятся к одной сети класса B с номером 129.34. Если задать в качестве маски число 255. 255.255, то адреса этих двух узлов будут интерпретироваться маршрутизаторами как адреса узла 15 сети класса C с номером 129.34.17 и узла 01 сети класса C с номером 129.34.20. Извне сеть по-прежнему будет выглядеть как единая сеть класса B, а на местном уровне это будет несколько отдельных сетей класса C.

Команда PING

Для проверки наличия в сети узла предназначена команда PING. Чтобы воспользоваться данной командой, необходимо выбрать пункт «выполнить» из меню «пуск». После чего ввести необходимую команду, например:

PING 192.168.0.1

PING rambler.ru

4.2. Протокол DHCP

Как уже было сказано, IP-адреса могут назначаться администратором сети вручную. Это представляет для администратора утомительную процедуру. Ситуация усложняется еще тем, что многие пользователи не обладают достаточными знаниями для того, чтобы конфигурировать свои компьютеры для работы в интерсети и должны поэтому полагаться на администраторов.

Протокол *Dynamic Host Configuration Protocol (DHCP)* был разработан для того, чтобы освободить администратора от этих проблем. Основным назначением DHCP является динамическое назначение IP-адресов. Однако, кроме динамического, DHCP может поддерживать и более простые способы ручного и автоматического статического назначения адресов.

В ручной процедуре назначения адресов активное участие принимает администратор, который предоставляет DHCP-серверу информацию о соответствии IP-адресов физическим адресам или другим идентификаторам клиентов. Эти адреса сообщаются клиентам в ответ на их запросы к DHCP-серверу.

При автоматическом статическом способе DHCP-сервер присваивает IP-адрес (и, возможно, другие параметры конфигурации клиента) из пула наличных IP-адресов без вмешательства оператора. Границы пула назначаемых адресов задает администратор при конфигурировании DHCP-сервера. Между идентификатором клиента и его IP-адресом по-прежнему,

как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первичного назначения сервером DHCP IP-адреса клиенту. При всех последующих запросах сервер возвращает тот же самый IP-адрес.

При динамическом распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, что дает возможность впоследствии повторно использовать IP-адреса другими компьютерами. Динамическое разделение адресов позволяет строить IP-сеть, количество узлов в которой намного превышает количество имеющихся в распоряжении администратора IP-адресов.

DHCP обеспечивает надежный и простой способ конфигурации сети TCP/IP, гарантируя отсутствие конфликтов адресов за счет централизованного управления их распределением. Администратор управляет процессом назначения адресов с помощью параметра "продолжительности аренды" (lease duration), которая определяет, как долго компьютер может использовать назначенный IP-адрес, перед тем как снова запросить его от сервера DHCP в аренду.

Примером работы протокола DHCP может служить ситуация, когда компьютер, являющийся клиентом DHCP, удаляется из подсети. При этом назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс. Это свойство очень важно для мобильных пользователей.

Протокол DHCP использует модель клиент-сервер. Во время старта системы компьютер-клиент DHCP, находящийся в состоянии "инициализация", посылает сообщение discover (исследовать), которое широковещательно распространяется по локальной сети и передается всем DHCP-серверам частной интерсети. Каждый DHCP-сервер, получивший это сообщение, отвечает на него сообщением offer (предложение), которое содержит IP-адрес и конфигурационную информацию.

Компьютер-клиент DHCP переходит в состояние "выбор" и собирает конфигурационные предложения от DHCP-серверов. Затем он выбирает одно из этих предложений, переходит в состояние "запрос" и отправляет сообщение request (запрос) тому DHCP-серверу, чье предложение было выбрано.

Выбранный DHCP-сервер посылает сообщение DHCP-acknowledgment (подтверждение), содержащее тот же IP-адрес, который уже был послан ранее на стадии исследования, а также параметр аренды для этого адреса. Кроме того, DHCP-сервер посылает параметры сетевой конфигурации. После того, как клиент получит это подтверждение, он переходит в состояние "связь", находясь в котором он может принимать участие в работе сети TCP/IP. Компьютеры-клиенты, которые имеют локальные диски, сохраняют полученный адрес для использования при последующих стартах системы. При приближении момента истечения срока аренды адреса, компьютер пытается обновить параметры аренды у DHCP-сервера, а если этот IP-адрес не может быть выделен снова, то ему возвращается другой IP-адрес.

В протоколе DHCP описывается несколько типов сообщений, которые используются для обнаружения и выбора DHCP-серверов, для запросов информации о конфигурации, для продления и досрочного прекращения лицензии на IP-адрес. Все эти операции направлены на то, чтобы освободить администратора сети от утомительных рутинных операций по конфигурированию сети.

Однако использование DHCP несет в себе и некоторые проблемы. Во-первых, это проблема согласования информационной адресной базы в службах DHCP и DNS. Как известно, DNS служит для преобразования символьных имен в IP-адреса. Если IP-адреса будут динамически изменяться сервером DHCP, то эти изменения необходимо также динамически вносить в базу данных сервера DNS.

Во-вторых, нестабильность IP-адресов усложняет процесс управления сетью. Системы управления, основанные на протоколе SNMP, разработаны с

расчетом на статичность IP-адресов. Аналогичные проблемы возникают и при конфигурировании фильтров маршрутизаторов, которые оперируют с IP-адресами.

Наконец, централизация процедуры назначения адресов снижает надежность системы: при отказе DHCP-сервера все его клиенты оказываются не в состоянии получить IP-адрес и другую информацию о конфигурации. Последствия такого отказа могут быть уменьшены путем использовании в сети нескольких серверов DHCP, каждый из которых имеет свой пул IP-адресов.

4.3. Протокол ICMP

ICMP (Internet Control Message Protocol) - протокол управляющих сообщений.

Компьютер получает их постоянно, а иногда и отправляет, например:

- Если адрес не доступен, вы получаете сообщение ICMP.
- Если порт не доступен, вы получаете сообщение ICMP.
- Если пользуетесь командой ping, вы получаете сообщение ICMP.
- и т.д.

Сообщение ICMP инкапсулируется прямо в IP- пакет (поле данных), т.е. протоколы транспортного уровня не используются.

Протокол ICMP представляет собой механизм передачи сообщений об ошибках, которые возникают в процессе информационного обмена в сети Internet. На данный протокол не возлагаются функции локализации и устранения причин, которые привели к возникновению этих ошибок.

Сообщения делятся на два типа:

- Парные (вопрос/ответ)
- Непарные (например: посылаете запрос к серверу, но сервер не доступен, и последний маршрутизатор (или сервер) отправляет ICMP-сообщение (Destination Unreachable) вам)

Сообщение Time Exceeded – (истекло время) принадлежит к непарным сообщениям ICMP. Это сообщение должно быть сформировано в том случае,

если в процессе передачи дейтаграммы истекло допустимое время её существования в сети или на хосте.

Непарное сообщение, формируется, если заголовок IP-дейтограммы содержит неверный параметр.

Непарное сообщение, формируется, если возникла перегрузка маршрутизатора, пакет не может быть помещен в буфер, так как он переполнен.

Непарное сообщение, формируется, если изменен маршрут для пакета.

Например (рис 4.3) хост A(10.40.0.2) отправляет дейтаграмму в направлении хоста B(10.10.0.2) используя для этого в качестве шлюза маршрутизатор R2. После того, как маршрутизатор R2 получает дейтаграмму, он определяет, что данная дейтаграмма адресована в направлении 10.10.0.0. Кратчайший маршрут для достижения этой сети для маршрутизатора R2 лежит через маршрутизатор R4, который в данном случае подключен к тому сегменту сети, из которого была получена принятая дейтаграмма.

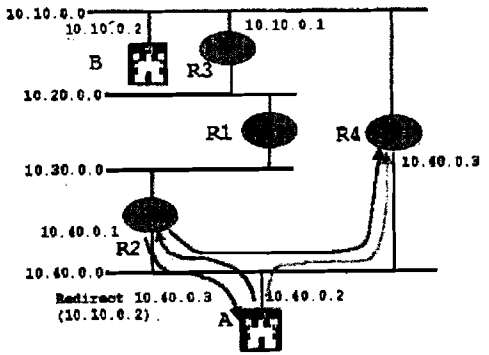


Рис. 4.3. Принцип работы ICMP

Маршрутизатор R2 направляет дейтаграмму по направлению R4 (красная стрелка на рисунке) и одновременно формирует сообщение ICMP Redirect, в котором он рекомендует хосту A впредь для передачи дейтаграмм в направлении сети использовать в качестве шлюза маршрутизатор R4.

4.4. Протокол EIGRP

EIGRP (Enhanced Interior Gateway Routing Protocol) – дистанционно векторный протокол маршрутизации [4], разработанный фирмой Cisco на основе протокола IGRP той же фирмы. Протокол IGRP был создан как альтернатива протоколу RIP (см. разд. 4.5), до того, как был разработан OSPF (см. разд. 4.6). После появления OSPF Cisco представила EIGRP – переработанный и улучшенный вариант IGRP, свободный от основного недостатка дистанционно-векторных протоколов – особых ситуаций с зацикливанием маршрутов – благодаря специальному алгоритму распространения информации об изменениях в топологии сети. Несмотря на то, что в общем случае протоколы состояния связей (OSPF) отрабатывают изменения в топологии сети быстрее, чем EIGRP, а также OSPF имеет ряд дополнительных возможностей, EIGRP более прост в реализации и менее требователен к вычислительным ресурсам маршрутизатора.

EIGRP-маршрутизатор обнаруживает своих соседей путем периодической рассылки сообщений "Hello". Эти же сообщения используются для мониторинга состояния связи с соседом (рассылаются каждые 5 секунд в сетях с большой пропускной способностью – например, Ethernet – и каждые 60 секунд в "медленных" сетях). Такой мониторинг позволяет рассылать в сети векторы расстояний не периодически, а только при изменении топологии сети.

EIGRP использует комплексное значение метрики (цены связи), вычисляемое на основании показателей пропускной способности и задержки при передаче данных в сети. Также в расчет метрики могут быть включены показатели загрузки и надежности сети. В отличие от протокола RIP метрика в EIGRP не является фактором, ограничивающим размер системы.

При получении от соседей векторов расстояний, маршрутизатор для каждой сети назначения не только выбирает соседа, через которого лежит кратчайший путь в эту сеть, но также запоминает и *вероятных заместителей* (*feasible successors*). Вероятным заместителем становится маршрутизатор,

объявивший метрику маршрута от себя до данной сети меньшую, чем полная метрика установленного маршрута. Рассмотрим пример на рис. 4.4 (для простоты метрики всех связей, кроме (4)-(5), считаем равными единице; метрика связи (4)-(5) равна 0,5).

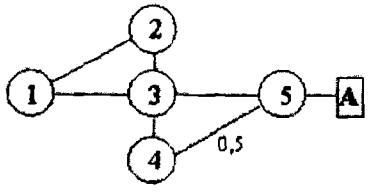


Рис 4.4. Пример EIGRP-системы

Кружками обозначены маршрутизаторы, прямоугольником – сеть назначения А. Маршрутизатор (3) получает от (5) элемент вектора расстояний (A=1), а от (4) – (A=1,5). В таблице маршрутизатора (3) узел (5) становится следующим маршрутизатором на пути в сеть А, а узел (4) – вероятным заместителем, так как заявленное им расстояние до А (1,5) меньше полной метрики установленного маршрута (3)-(5)-А, которая равна 2.

Обратим внимание на маршрутизаторы (1) и (2). Они присылают узлу (3) элемент (A=3) и, следовательно, не являются вероятными заместителями маршрута из (3) в А, что, безусловно, разумно.

Если связь между узлами (3) и (5) обрывается, то (3) ищет в своей EIGRP-таблице вероятного заместителя (4) и немедленно устанавливает маршрут в сеть А через него. Таким образом время, в течение которого маршрут в сеть А отсутствовал, существенно сокращается по сравнению с протоколами, где требуется ждать, когда соседи пришлют очередные векторы расстояний.

Если же ни одного вероятного заместителя не найдено (допустим, связь (3)-(4) тоже обрывается), то маршрутизатор переходит в активное состояние и начинает опрос всех своих соседей на предмет наличия маршрута в сеть А, сообщая при этом что его собственное расстояние до А равно бесконечности.

Сосед отвечает на запрос только тогда, когда у него есть либо готовый маршрут в А, либо вероятный заместитель – в любом из этих случаев сосед присылает в узел (3) свое расстояние до А. Иначе сосед сам переходит в активное состояние и процесс повторяется (разумеется, с той разницей, что к маршрутизатору (3) запрос не посылается; кроме того, маршрутизатор, находящийся в активном состоянии, сам может отвечать на запросы, посылая в ответ свое текущее значение расстояния до А). Таким образом область "активизированных" маршрутизаторов расширяется до тех пор, пока не будет обнаружен маршрут в сеть А или доказано его отсутствие, после чего волна сходится в обратном направлении к инициировавшему процесс узлу, при этом все маршрутизаторы вносят в свои таблицы надлежащие изменения.

В этом простом примере, после того как (3) переходит в активное состояние, узлы (1) и (2) получают от него запрос о маршруте в сеть А с пометкой, что расстояние от (3) до А теперь равно бесконечности. Каждый из них, поскольку ранее он добирался в А через (3), помечает этот маршрут как недостижимый, и, не найдя вероятного заместителя, активизируется и опрашивает своего соседа. Получив эти запросы, (1) и (2) отвечают друг другу, что сеть А недостижима, переходят в пассивное состояние и возвращают узлу (3) информацию о недостижимости сети А.

4.5. Протокол RIP

Протокол RIP является дистанционно-векторным протоколом внутренней маршрутизации [4]. Процесс работы протокола состоит в рассылке, получении и обработке векторов расстояний до IP-сетей, находящихся в области действия протокола, то есть в данной RIP-системе. Результатом работы протокола на конкретном маршрутизаторе является таблица, где для каждой сети данной RIP-системы указано расстояние до этой сети (в хопах) и адрес следующего маршрутизатора.

Алгоритм построения таблицы маршрутов

В этом разделе для простоты будем называть таблицей маршрутов таблицу, являющуюся результатом деятельности протокола RIP, как описано

выше, т.е. состоящую из строк с полями "Сеть", "Расстояние", "Следующий маршрутизатор". Записывать строку в таблице маршрутов будем следующим образом:

A=2→③

Это означает, что расстояние от данного маршрутизатора до сети A равно 2, а дейтаграммы, следующие в сеть A, надо пересылать маршрутизатору.

Вектором расстояний называется пара ("Сеть", "Расстояние до этой сети"), извлеченная из таблицы маршрутов. Каждую такую пару называют элементом вектора расстояний. Будем записывать вектор расстояний в виде (A=2,B=1): это означает, что расстояние от данного маршрутизатора до сети A равно 2, до сети B равно 1.

Расстояние до сети, к которой маршрутизатор подключен непосредственно, примем равным 1.

Каждый маршрутизатор, на котором запущен модуль RIP, периодически широковещательно распространяет свой вектор расстояний. Вектор распространяется через все интерфейсы маршрутизатора, подключенные к сетям, входящим в RIP-систему.

Каждый маршрутизатор также периодически получает векторы расстояний от других маршрутизаторов. Расстояния в этих векторах увеличиваются на 1, после чего сравниваются с данными в таблице маршрутов, и, если расстояние до какой-то из сетей в полученном векторе оказывается меньше расстояния, указанного в таблице, значение из таблицы замещается новым (меньшим) значением, а адрес маршрутизатора, приславшего вектор с этим значением, записывается в поле "Следующий маршрутизатор" в этой строке таблицы. После этого вектор расстояний, рассылаемый данным маршрутизатором, соответственно изменится.

Пример построения таблицы маршрутов

Рассмотрим этот процесс на примере следующей сети (рис 4.5).

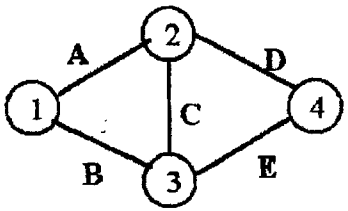


Рис. 4.5. Пример RIP-системы

Здесь ①, ②, ③, ④ - маршрутизаторы, A, B, C, D, E - сети. Хосты в сетях не показаны за ненадобностью. Мы будем следить за формированием таблицы маршрутов в узле ①.

В начальный момент времени (например, после подачи питания на маршрутизаторы) таблица маршрутов в узле ① выглядит следующим образом (т.к. узел ① знает только о тех сетях, к которым подключен непосредственно):

A=1→①
B=1→①

Следовательно, узел ① рассылает в сети A и B вектор расстояний (A=1,B=1).

Аналогично узел ② рассылает в сети A, C, D вектор (A=1,C=1,D=1). Узел ① получает этот вектор из сети A, увеличивает расстояния на 1 (A=2,C=2,D=2) и сравнивает с данными в своей таблице маршрутов. Новое расстояние до сети A оказывается больше, чем уже внесенное в таблицу (A=1), следовательно, новое значение игнорируется. Поскольку сети C и D вовсе не фигурируют в его таблице маршрутов, они туда вносятся. В узле ① имеем:

A=1→①
B=1→①
C=2→②
D=2→②

Узел ④ в свою очередь рассылает вектор (D=1,E=1) в сети D и E. Узел ② получает этот вектор из сети D, увеличивает расстояния на 1, после чего добавляет себе в таблицу данные о сети E (E=2→②). Ранее из узла ① он получил информацию о сети B и добавил себе в таблицу строку B=2→①. Узел ② рассылает в сети A, C, D свой обновленный вектор расстояний (A=1,B=2,C=1,D=1,E=2).

Узел ① получает этот вектор от ② из сети A, увеличивает расстояния на 1: (A=2,B=3,C=2,D=2,E=3) и замечает, что все указанные расстояния, кроме расстояния до сети E, больше, либо равны значениям, имеющимся в его таблице. Сеть E в таблице узла ① отсутствует, следовательно, она туда вносится, и в узле ① мы получаем:

A=1→①
B=1→①
C=2→②
D=2→②
E=3→②

Далее маршрутизатор ③ ранее не работавший по каким-либо причинам, рассылает в сети B, C, E свой вектор (B=1,C=1,E=1). Узел ① получает этот вектор из сети B, увеличивает расстояния на 1 и обнаруживает, что расстояние E=2 меньше имеющегося в таблице E=3, следовательно запись о сети E в таблице заменяется на E=2→③. Остальные элементы полученного от ③ вектора не вызывают обновления таблицы.

Итоговая таблица маршрутов маршрутизатора ① :

A=1→①
B=1→①
C=2→②
D=2→②
E=2→③

На этом алгоритм сходится, то есть при неизменной топологии системы никакие векторы расстояний, получаемые маршрутизатором ①, больше не внесут изменений в таблицу маршрутов. Аналогичным образом алгоритм

составления таблицы маршрутов работает и сходится на других маршрутизаторах. Отметим, что несмотря на то, что таблицы маршрутов построены, векторы расстояний продолжают периодически широковещательно рассылаться каждым маршрутизатором. Это требуется для оперативного реагирования на внезапные изменения топологии системы.

Очевидно, что вид построенной таблицы маршрутов может зависеть от порядка получения маршрутизатором векторов расстояний. Например, если бы узел ① получил вектор от узла ③ раньше, чем от узла ②, то дейтаграммы в сеть C посылались бы от ① через ③.

Изменение состояния RIP-системы

Выясним, что происходит в случае, когда состояние системы неожиданно изменяется, например, маршрутизатор ① отключается от сети A (рис 4.6).

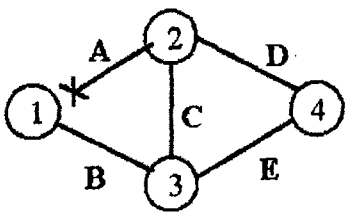


Рис. 4.6. Изменение состояния RIP-системы

Узел ① обнаруживает свое отсоединение от сети A и меняет таблицу маршрутов, устанавливая бесконечное расстояние до всех сетей, ранее достижимых через маршрутизаторы, подключенные к сети A (то есть ②). В протоколе RIP значение бесконечности равно 16.

A=16→①
B=1→①
C=16→②
D=16→②
E=2→③

Вектор расстояний, построенный на основании этой таблицы, рассылается в сеть B, чтобы маршрутизаторы, направлявшие свои данные

через ① в ставшие недоступными сети, если таковые маршрутизаторы существуют, соответственно изменили свои маршрутные таблицы.

Допустим, в узле ③ имелась следующая таблица маршрутов:

A=2→②
B=1→③
C=1→③
D=2→④
E=1→③

Узел ③ периодически и широковещательно рассылает в сети B, C, E свой вектор расстояний (A=2,B=1,C=1,D=2,E=1). Узел ① получает этот вектор, увеличивает расстояния на 1: (A=3,B=2,C=2,D=3,E=2) и замечает, что расстояния A=3, C=2 и D=3 меньше бесконечности следовательно, соответствующие записи таблицы маршрутов модифицируются и она принимает вид:

A=3→③
B=1→①
C=2→③
D=3→③
E=2→③

Таким образом, узел ① построил маршруты в обход поврежденного участка и восстановил достижимость всех сетей.

Особые случаи

Зацикливание

К сожалению, поведение дистанционно-векторных протоколов (и в частности, протокола RIP) при изменении топологии системы не всегда корректно и предсказуемо.

Рассмотрим вышеописанную ситуацию с отсоединением узла ① от сети A (рис 4.7).

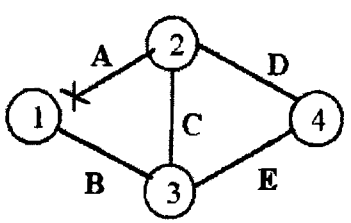


Рис. 4.7. Изменение состояния RIP-системы

Мы предполагали, что узел ③ не отправлял дейтаграмм через узел ① (и, следовательно, изменение таблицы маршрутов в узле ① не повлияло на таблицу узла ③). Предположим теперь, что ③ отправлял дейтаграммы в сеть A через ① , то есть таблица в узле ③ имела вид:

A=2→①
B=1→③
C=1→③
D=2→④
E=1→③

После отсоединения ① от сети A узел ③ получает от ① вектор (A=16,B=1,C=16,D=16,E=2). Проанализировав этот вектор, ③ делает вывод, что все указанные в нем расстояния больше значений, содержащихся в его маршрутной таблице, на основании чего этот вектор узлом ③ игнорируется.

В свою очередь узел ③ рассылает в сети B, C, E вектор (A=2,B=1,C=1,D=2,E=1). Узел ① получает этот вектор, увеличивает расстояния на 1: (A=3,B=2,C=2,D=3,E=2) и замечает, что расстояния A=3, C=2 и D=3 меньше бесконечности, следовательно, соответствующие записи таблицы маршрутов в узле ① модифицируются и она принимает вид:

A=3→③
B=1→①
C=2→③
D=3→③
E=2→③

Очевидно, после этого содержимое таблиц узлов ① и ③ стабилизируется.

Рассмотрим теперь записи о достижении сети A в таблицах маршрутизаторов ① и ③ .

В узле ①: A=3→③
В узле ③: A=2→①

Таким образом, возникло зацикливание: данные, адресованные в сеть A, будут пересылаться между узлами ① и ③ до тех пор, пока не истечет время жизни дейтаграмм и они не будут уничтожены.

Для того, чтобы избежать зацикливания, в алгоритм рассылки векторов расстояний вносятся дополнения. Тем не менее и в этом случае особые ситуации все еще остаются.

4.6. Протокол OSPF

Протокол маршрутизации OSFP (Open Shortest Path First) представляет собой протокол состояния связей, использующий алгоритм SPF поиска

кратчайшего пути в графе [4]. OSPF применяется для внутренней маршрутизации в системах сетей любой сложности.

Построение маршрутов

Рассмотрим работу алгоритма SPF и построение маршрутов на примере системы, изображенной на рис. 4.8. Для простоты будем рассматривать OSPF-систему, состоящую только из маршрутизаторов, соединенных линиями связи типа "точка-точка".

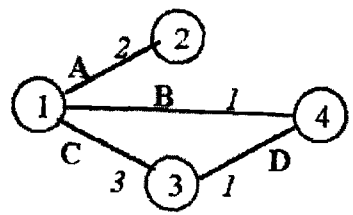


Рис. 4.8. Пример OSPF-системы:
①, ②, ③, ④ - маршрутизаторы; A,B,C,D - линии связи (или просто связи), цифры означают метрику каждой связи

Метрики

Метрика представляет собой оценку качества связи в данной сети (на данном физическом канале); чем меньше метрика, тем лучше качество соединения. Метрика маршрута равна сумме метрик всех связей (сетей), входящих в маршрут. В простейшем случае (как это имеет место в протоколе RIP) метрика каждой сети равна единице, а метрика маршрута тогда просто является его длиной в хопах.

Метрика сети, оценивающая пропускную способность, определяется как количество секунд, требуемое для передачи 100 Мбит через физическую среду данной сети. Например, метрика сети на базе 10Base-T Ethernet равна 10, а метрика выделенной линии 56 кбит/с равна 1785. Метрика канала со скоростью передачи данных 100 Мбит/с и выше равна единице.

Порядок расчета метрик, оценивающих надежность, задержку и стоимость, не определен. Администратор, желающий поддерживать маршрутизацию по этим типам сервисов, должен сам назначить разумные и согласованные метрики по этим параметрам.

В нашем примере мы будем использовать метрики, указанные на рисунке, без учета типов сервиса. Следует заметить, что маршрутизация по типам сервиса крайне редка, более того, она исключена из последних версий стандарта OSPF.

База данных состояния связей

Для работы алгоритма SPF на каждом маршрутизаторе строится база данных состояния связей. Базы данных на всех маршрутизаторах идентичны.

База данных состояния связей для рассмотренного примера приведена в табл. 4.2.

База данных состояния связей Таблица 4.2

От → До	Сеть	Метрика
①→②	A	2
①→③	C	3
①→④	B	1
②→①	A	2
③→①	C	3
③→④	D	1
④→①	B	1
④→③	D	1

5. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ

С появлением вычислительных сетей возникла необходимость в разработке концепции, устанавливающей стандартные правила взаимодействия разнотипных станций (узлов) входящих в состав сети. Для этого была создана многоуровневая архитектура с выделением 7 уровней иерархии. Разработчики – ряд международных организаций по стандартизации ISO, IEEE и др. назвали созданную ими в начале 80-х годов структуру *моделью взаимодействия открытых систем* (Open System Interconnection, OSI). Согласно модели OSI, функции взаимодействия объектов в сети разделены на следующие уровни: физический, канальный, сетевой, транспортный, сеансовый, представительный и прикладной.

Модель OSI описывает только системные функции, реализуемые операционной системой, системными сетевыми компонентами, системными аппаратными средствами и не включает средства взаимодействия приложений конечных пользователей.

Приложения конечных пользователей взаимодействуют, обращаясь к системным средствам.

Поскольку в процессе обмена сообщениями в сети участвуют две стороны, то необходимо организовать согласованную работу двух «иерархий», работающих на разных узлах. Оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения размера сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты для всех уровней, начиная от самого низкого - уровня передачи битов - до самого высокого, реализующего сервис для пользователей сети.

В качестве примера на рис. 5.1 показана модель взаимодействия двух узлов. С каждой стороны средства взаимодействия представлены четырьмя уровнями. Процедура взаимодействия этих двух узлов может быть описана в виде набора правил взаимодействия каждой пары соответствующих уровней

обеих участвующих сторон. Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются *протоколом.*

Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другом в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений. Эти правила принято называть *интерфейсом.* Интерфейс определяет набор сервисов, предоставляемый данным уровнем соседнему уровню. В сущности, протокол и интерфейс выражают одно и то же понятие, но традиционно в сетях за ними закрепили разные области действия.

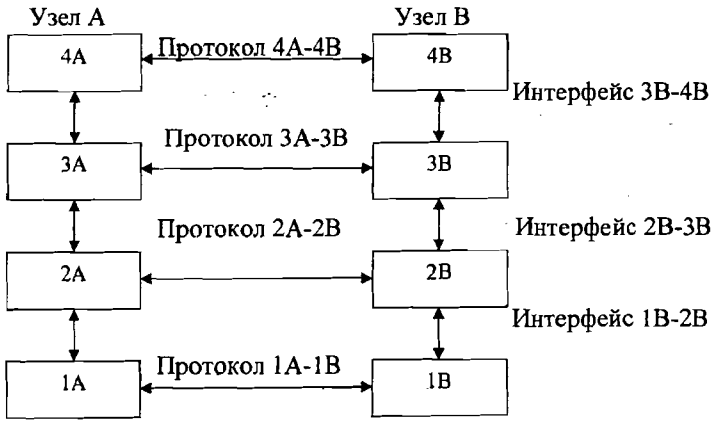


Рис.5.1. Иллюстрация места протоколов и интерфейсов в иерархии взаимодействия открытых систем

Протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы - правила взаимодействия модулей соседних уровней в одном узле. Средства каждого уровня должны отрабатывать, во-первых, свой собственный протокол, а во-вторых, интерфейсы с соседними уровнями.

Иерархически организованный набор протоколов, достаточный

для организации взаимодействия узлов в сети, называется *стеком коммуникационных протоколов*.

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней - как правило, чисто программными средствами.

Протоколы реализуются различными сетевыми устройствами - концентраторами, мостами, коммутаторами, маршрутизаторами и пр. Действительно, в общем случае связь узлов в сети осуществляется не напрямую, а через различные коммуникационные устройства.

5.1. Уровни модели OSI

Физический уровень

Физический уровень (Physical layer) имеет дело с передачей битов по физическим каналам связи, таким, например, как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. К этому уровню имеют отношение характеристики физических сред передачи данных, такие, как полоса пропускания, помехозащищенность, волновое сопротивление и др. На этом же уровне определяются характеристики электрических сигналов, передающих дискретную информацию, например крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизуются типы разъемов и назначение каждого контакта [5].

Функции физического уровня реализуются во всех устройствах, подключенных к сети и выполняются сетевым адаптером или последовательным портом. Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 м, манчестерский код для представления данных в

кабеле, а также некоторые другие характеристики среды и электрических сигналов.

Канальный уровень

На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих станций, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня (Data Link Layer) является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (frames). Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность битов в начало и конец каждого кадра для его выделения, а также вычисляет контрольную сумму, обрабатывая все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит по сети, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка. Канальный уровень может не только обнаруживать ошибки, но и исправлять их за счет повторной передачи поврежденных кадров. Необходимо отметить, что функция исправления ошибок не является обязательной для канального уровня, поэтому в некоторых протоколах этого уровня она отсутствует, например в Ethernet [5].

Хотя канальный уровень и обеспечивает доставку кадра между любыми двумя узлами локальной сети, он это делает только в сети с совершенно определенной топологией связей, именно той топологией, для которой он был разработан. К таким типовым топологиям, поддерживаемым протоколами канального уровня локальных сетей, относятся общая шина, кольцо и звезда, а также структуры, полученные из них с помощью мостов и

коммутаторов. Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI.

В локальных сетях протоколы канального уровня используются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

Для обеспечения качественной транспортировки сообщений в сетях любых топологий и технологий функций канального уровня оказывается недостаточно, поэтому в модели OSI решение этой задачи возлагается на два следующих уровня - сетевой и транспортный.

Канальный уровень обеспечивает передачу пакетов данных, поступающих от протоколов верхних уровней, узлу назначения, адрес которого также указывает протокол верхнего уровня. Протоколы канального уровня оформляют переданные им пакеты в кадры собственного формата, помещая указанный адрес назначения в одно из полей такого кадра, а также сопровождая кадр контрольной суммой. Протокол канального уровня имеет локальный смысл, он предназначен для доставки кадров данных, как правило, в пределах сетей с простой топологией связей и однотипной или близкой технологией, например в односегментных сетях Ethernet или же в многосегментных сетях Ethernet и Token Ring иерархической топологии, разделенных только мостами и коммутаторами. Во всех этих сетях адрес назначения имеет локальный смысл для данной сети и не изменяется при прохождении кадра от узла источника к узлу назначения.

Сетевой уровень

Сетевой уровень (Network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать совершенно разные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей. Функции сетевого уровня достаточно разнообразны. Протоколы канального уровня локальных сетей обеспечивают доставку данных между любыми

узлами только в сети с соответствующей типовой топологией, например топологией иерархической звезды. Это очень жесткое ограничение, которое не позволяет строить сети с развитой структурой, например сети, объединяющей несколько сетей предприятия в единую сеть, или высоконадежные сети, в которых существуют избыточные связи между узлами. Чтобы, с одной стороны, сохранить простоту процедур передачи данных для типовых топологий, а с другой - допустить использование произвольных топологий, вводится дополнительный сетевой уровень [5].

Внутри сети доставка данных обеспечивается соответствующим канальным уровнем, а доставкой данных между сетями занимается сетевой уровень, который и поддерживает возможность правильного выбора маршрута передачи сообщения даже в том случае, когда характер структуры связей между составляющими сетями отличается от принятого в протоколах канального уровня.

Сети соединяются между собой специальными устройствами, называемыми маршрутизаторами. ***Маршрутизатор*** - это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. Чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач между сетями, каждый раз выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет. **Проблема выбора наилучшего пути называется *маршрутизацией,* и ее решение является одной из главных задач сетевого уровня.** Эта проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени. Некоторые алгоритмы маршрутизации пытаются приспособиться к изменению нагрузки, в то время

как другие принимают решения на основе средних показателей за длительное время. Выбор маршрута может осуществляться и по другим критериям, например надежности передачи. В общем случае функции сетевого уровня шире, чем функции передачи сообщений по связям с нестандартной структурой. Сетевой уровень решает также задачи согласования разных технологий, упрощения адресации в крупных сетях и создания надежных и гибких барьеров на пути нежелательного трафика между сетями.

Сообщения сетевого уровня принято называть ***пакетами*** (packet). При организации доставки пакетов на сетевом уровне используется понятие «номер сети». В этом случае адрес получателя состоит из старшей части - номера сети и младшей - номера узла в этой сети. Все узлы одной сети должны иметь одну и ту же старшую часть адреса, поэтому термину «сеть» на сетевом уровне можно дать и другое, более формальное определение: сеть - это совокупность узлов, сетевой адрес которых содержит один и тот же номер сети.

На сетевом уровне определяются два вида протоколов. Первый вид - ***сетевые протоколы*** - реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых протоколами обмена маршрутной информацией или просто ***протоколами маршрутизации***. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений. Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов. На сетевом уровне работают протоколы еще одного типа, которые отвечают за отображение адреса узла, используемого на сетевом уровне, в локальный адрес сети. Такие протоколы часто называют ***протоколами разрешения адресов***. Иногда их относят не к сетевому уровню, а к канальному уровню.

Примерами протоколов сетевого уровня являются протокол

межсетевого взаимодействия IP стека TCP /IP и протокол межсетевого обмена пакетами IPX стека Novell.

Транспортный уровень

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Транспортный уровень (Transport layer) обеспечивает приложениям или верхним уровням стека - прикладному и сеансовому - передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное - способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов [5].

Выбор класса сервиса транспортного уровня определяется, с одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких, чем транспортный, уровней, а с другой стороны, этот выбор зависит от того, насколько надежной является система транспортировки данных в сети, обеспечиваемая уровнями, расположенными ниже транспортного - сетевым, канальным и физическим. Так, например, если качество каналов передачи связи очень высокое и вероятность возникновения ошибок, не обнаруженных протоколами более низких уровней, невелика, то разумно воспользоваться одним из облегченных сервисов транспортного уровня, не обремененных многочисленными проверками, квитированием и другими приемами повышения надежности. Если же транспортные средства нижних уровней изначально ненадежны, то целесообразно обратиться к наиболее развитому

сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок, включая предварительное установление логического соединения, контроль доставки сообщений по контрольным суммами циклической нумерации пакетов, установление тайм-аутов доставки и т. п

Протоколы нижних четырех уровней обобщенно называют сетевым транспортом или транспортной подсистемой, так как они полностью решают задачу транспортировки сообщений с заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями. Оставшиеся три верхних уровня решают задачи предоставления прикладных сервисов на основании имеющейся транспортной подсистемы.

Сеансовый уровень

Сеансовый уровень (Session layer) обеспечивает управление взаимодействием станций в сети: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов, хотя функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе [5].

Представительный уровень

Представительный уровень (Presentation layer) имеет дело с формой представления передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC. На

этом уровне может выполняться шифрование и дешифрование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол Secure Socket Layer (SSL), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

Прикладной уровень

Прикладной уровень (Application layer) - это просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые web-страницы и т.д. Единица данных, которой оперирует прикладной уровень, называется ***сообщением*** *(message).*

Сетезависимые и сетенезависимые уровни

Функции всех уровней модели OSI могут быть отнесены к одной из двух групп: либо к функциям, зависящим от конкретной технической реализации сети, либо к функциям, ориентированным на работу с приложениями.

Три нижних уровня - физический, канальный и сетевой - являются сетезависимыми, то есть протоколы этих уровней тесно связаны с технической реализацией сети и используемым коммуникационным оборудованием. Например, переход на оборудование FDDI означает полную смену протоколов физического и канального уровней во всех узлах сети.

Три верхних уровня - прикладной, представительный и сеансовый - ориентированы на приложения и мало зависят от технических особенностей построения сети. На протоколы этих уровней не влияют какие бы то ни было изменения в топологии сети: замена оборудования или переход на другую сетевую технологию. Транспортный уровень является промежуточным, он скрывает детали функционирования нижних уровней от верхних. Это позволяет разрабатывать приложения, не зависящие от технических средств транспортировки приложений.

5.2. Стек протоколов TCP/IP

TCP/IP - собирательное название для набора (стека) сетевых протоколов разных уровней, используемых в INTERNET. Особенности TCP/IP:

- открытые стандарты протоколов, разрабатываемые независимо от программного и аппаратного обеспечения;
- независимость от физической среды передачи;
- система уникальной адресации;
- стандартизованные протоколы высокого уровня для распространенных пользовательских сервисов.

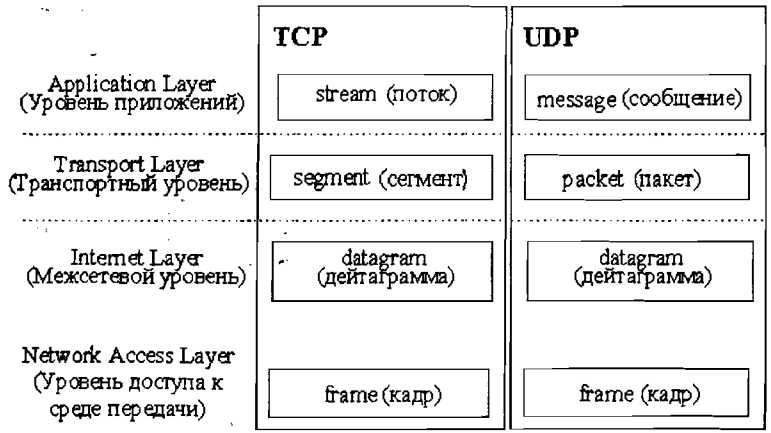


Рис. 5.2. Стек протоколов TCP/IP

Стек протоколов TCP/IP (рис. 5.2) делится на 4 уровня: ***прикладной*** *(application)*, ***транспортный*** *(transport)*, ***межсетевой*** *(internet)* и ***уровень доступа к среде передачи*** *(network access)*. Термины, применяемые для обозначения блока передаваемых данных, различны при использовании разных протоколов транспортного уровня - TCP и UDP, поэтому на рисунке изображено два стека. Как и в модели OSI, данные более верхних уровней инкапсулируются в пакеты нижних уровней (рис. 5.3).

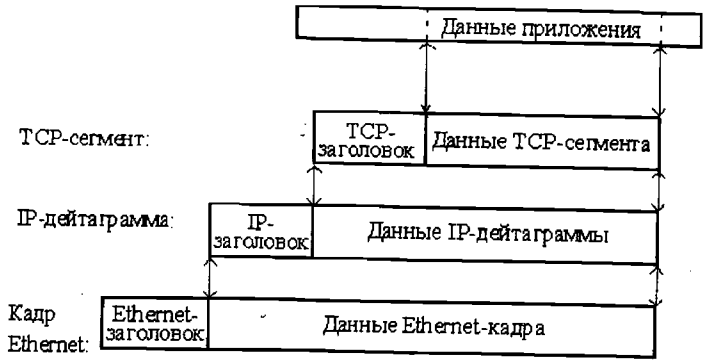


Рис. 5.3. Пример инкапсуляции пакетов в стеке TCP/IP

Примерное соотношение уровней стеков OSI и TCP/IP показано на рис. 5.4.

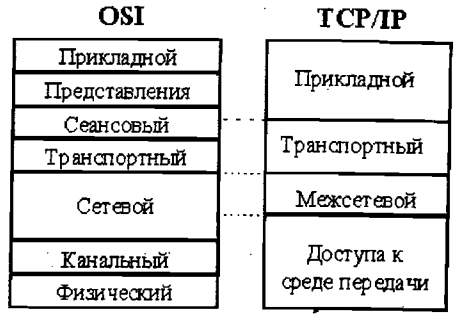


Рис. 5.4. Соотношение уровней стеков OSI и TCP/IP

Ниже кратко рассматриваются функции каждого:

Прикладной уровень (уровень приложений)

Приложения, работающие со стеком TCP/IP, могут также выполнять функции уровней представления и частично сеансового модели OSI; например, преобразование данных к внешнему представлению, группировка данных для передачи и т.п.

Примерами приложений являются HTTP-серверы и клиенты (WWW-броузеры), программы работы с электронной почтой.

Для пересылки данных другому приложению, приложение обращается к тому или иному модулю транспортного уровня.

Транспортный уровень

Протоколы транспортного уровня обеспечивают прозрачную (сквозную) доставку данных (end-to-end delivery service) между двумя прикладными процессами. Процесс, получающий или отправляющий данные с помощью транспортного уровня, идентифицируется на этом уровне номером, который называется номером порта. Таким образом, роль адреса отправителя и получателя на транспортном уровне выполняет номер порта (или проще - порт).

Анализируя заголовок своего пакета, полученного от межсетевого уровня, транспортный модуль определяет по номеру порта получателя, какому из прикладных процессов направлены данные, и передает эти данные соответствующему прикладному процессу (возможно, после проверки их на наличие ошибок и т.п.). Номера портов получателя и отправителя записываются в заголовок транспортным модулем, отправляющим данные; заголовок транспортного уровня содержит также и другую служебную информацию; формат заголовка зависит от используемого транспортного протокола.

На транспортном уровне работают два основных протокола: UDP и TCP.

Межсетевой уровень

Основным протоколом этого уровня является протокол IP (Internet Protocol).

Протокол IP доставляет блоки данных, называемых дейтаграммами, от одного IP-адреса к другому. IP-адрес является уникальным 32-битным идентификатором компьютера (точнее, его сетевого интерфейса). Данные для дейтаграммы передаются IP-модулю транспортным уровнем. IP-модуль предваряет эти данные заголовком, содержащим IP-адреса отправителя и

получателя и другую служебную информацию, и сформированная таким образом дейтаграмма передается на уровень доступа к среде передачи (например, одному из физических интерфейсов) для отправки по каналу передачи данных.

Не все компьютеры могут непосредственно связаться друг с другом; часто для того, чтобы передать дейтаграмму по назначению, требуется направить ее через один или несколько промежуточных компьютеров по тому или иному маршруту. Задача определения маршрута для каждой дейтаграммы решается протоколом IP.

Уровень доступа к среде передачи

Функции этого уровня:

- отображение IP-адресов в физические адреса сети (MAC-адреса, например, Ethernet-адрес в случае сети Ethernet). Эту функцию выполняет протокол ARP (см. раздел 3.5);
- инкапсуляция IP-дейтаграмм в кадры для передачи по физическому каналу и извлечение дейтаграмм из кадров. При этом не требуется какого-либо контроля безошибочности передачи (хотя он может и присутствовать), поскольку в стеке TCP/IP такой контроль возложен на транспортный уровень или на само приложение;
- определение метода доступа к среде передачи - то есть способа, с помощью которого компьютер устанавливает свое право на произведение передачи данных;
- определение представления данных в физической среде;
- пересылка и прием кадра.

6. ПРОМЫШЛЕННЫЕ СЕТИ

6.1. HART – протокол

Полевой коммуникационный протокол HART широко применяется в промышленности как стандарт для цифровой коммуникации со "smart"-приборами. Протокол HART (Highway Addressable Remote Transducer) разработан фирмой Rosemount Inc. в середине 80-х годов, реализует известный стандарт BELL 202 FSK (Frequency Shift Keying).

Его особенность в том, что он использует для передачи цифровых данных низкоуровневую частотную модуляцию, наложенную на аналоговый сигнал 4-20 mA (токовая петля). Обмен данными по HART протоколу происходит на скорости 1200 Бод. Эта единица названа в честь Эмиля Бодо (Jean Maurice-Emile Baudot) (1845-1903), французского инженера по телеграфии, изобретателя первого печатающего устройства для телеграфа. Схема, поясняющая работу приборов по HART протоколу, представлена на рис.6.1.

Для передачи логической "1" HART использует один полный период частоты 1200 Гц, а для передачи логического "0" - два неполных периода 2200 Гц. Как видно на рис.6.1, HART составляющая накладывается на токовую петлю 4-20 мА. Поскольку среднее значение синусоиды за период равно "0", то HART сигнал никак не влияет на аналоговый сигнал 4-20 мА, который поэтому тоже может использоваться.

HART протокол построен по принципу "главный - подчиненный", то есть полевое устройство отвечает по запросу системы. Протокол допускает наличие двух управляющих устройств (управляющая система и коммуникатор).

Существует два режима работы датчиков, поддерживающих обмен данными по HART протоколу.

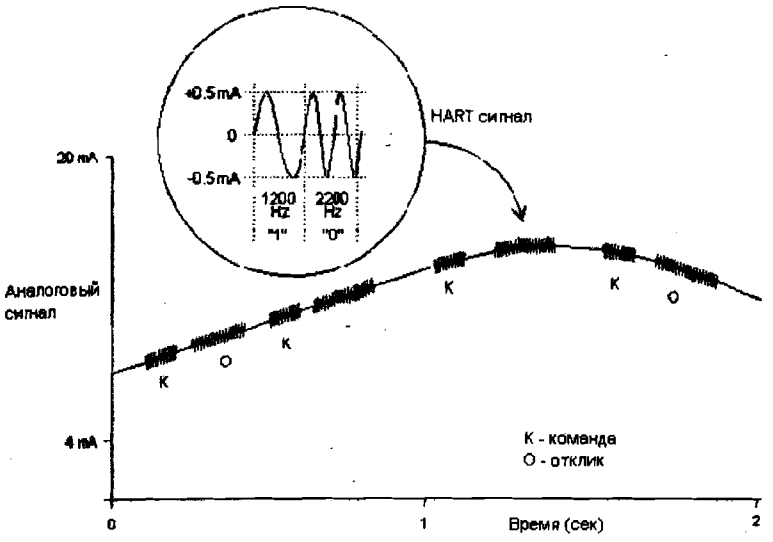


Рис.6.1. Схема работы HART протокола

6.2. Одноточечный режим передачи данных

Режим передачи цифровой информации одновременно с аналоговым сигналом представлен на рис.6.2. Обычно в этом режиме датчик работает в аналоговых АСУ ТП, а обмен по HART-протоколу осуществляется посредством HART коммуникатора или компьютера. При этом можно удаленно (расстояние до 3000 м) осуществлять полную настройку и конфигурирование датчика. Теперь оператору нет необходимости обходить все датчики на предприятии, он может их настроить непосредственно со своего рабочего места.

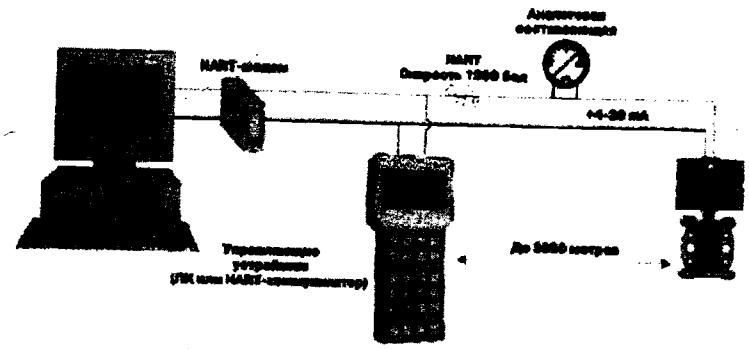


Рис.6.2. Одноточечный режим передачи цифровой информации

6.3. Многоточечный режим передачи данных

В многоточечном режиме (рис.6.3) датчик передает и получает информацию только в цифровом виде. Аналоговый выход автоматически фиксируется на минимальном значении (только питание устройства - 4 мА) и не содержит информации об измеряемой величине. Информация о переменных процесса считывается по HART-протоколу.

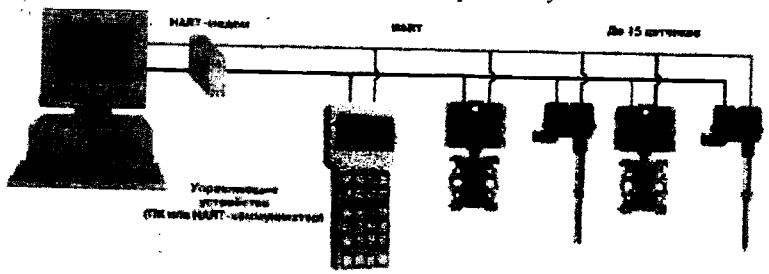


Рис.6.3. Многоточечный режим передачи информации

К одной паре проводов может быть подключено до 15 датчиков. Их количество определяется длиной и качеством линии, а так же мощностью блока питания датчиков. Все датчики в многоточечном режиме имеют свой уникальный адрес от 1 до 15, и обращение к каждому идет по соответствующему адресу. Коммуникатор или система управления определяет все датчики, подключенные к линии, и может работать с любым из них.

Обычно в аналоговой АСУТП присутствует множество интеллектуальных полевых приборов, работающих в режиме 4-20мА + HART. В этом случае удаленная настройка и конфигурирование датчиков при помощи HART-коммуникатора или HART-модема требует последовательного подключения коммуникационного устройства к каждой линии 4-20 мА, идущей от соответствующих приборов. Для решения поставленной задачи предлагается использовать HART-мультиплексор. При таком подходе приборы продолжают передавать измерительную информацию в систему по токовому выходу 4-20 мА, а их конфигурация может быть изменена с одного цифрового выхода управляющей системы. Связь мультиплексора с системой управления осуществляется по интерфейсу RS485 или RS232. При этом можно объединить в сеть около 500 приборов (например, 30 мультиплексоров соединенных по RS485, 16 каналов каждый). Структурная схема работы мультиплексора в аналоговой системе представлена на рис.6.4 (линии 2,3,..n).

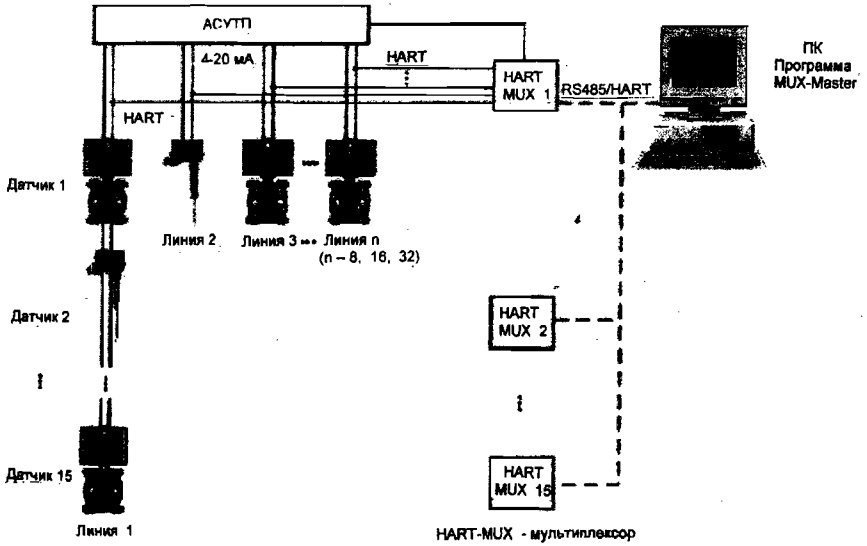


Рис.6.4. Структурная схема работы мультиплексора

Существует возможность построения с помощью мультиплексора цифровой системы сбора и визуализации информации. В этом случае

каждый канал мультиплексора может опрашивать до 15 датчиков, подключенных к одной токовой петле. При таком подключении затраты на кабельную продукцию существенно снижаются (рис. 6.4).

6.4. Определение промышленной сети

Промышленная сеть — это набор стандартных протоколов обмена данными, позволяющих связать воедино оборудование (датчики, исполнительные механизмы, промышленные контроллеры) различных производителей, а также обеспечить взаимодействие нижнего и верхнего уровней АСУ.

В промышленных сетях для передачи данных применяют кабели, оптоволоконные линии, беспроводную связь. Промышленные сети могут взаимодействовать с обычными компьютерными сетями, в частности использовать глобальную сеть Internet.

Выделяют три наиболее значимых параметра, по которым классифицируют сети [4]: топология сети, объем информационного сервиса, предоставляемого сетью, и способ доступа к физическому каналу передачи данных.

Топология сетей (общая шина, кольцо, звезда) рассмотрена в 1.7.

6.5. Объем информационного сервиса

О модели (OSI), разграничивающей и формализующей функции, выполняемые различными уровнями аппаратного и программного обеспечения сетевой структуры, рассказано в разд. 5. На практике большинство промышленных сетей ограничивается только тремя уровнями: физическим, канальным и прикладным. Дешевые сети (например, ModBus) зачастую используют на физическом уровне интерфейс RS-232 или RS-485, а все остальные задачи, начиная с канального уровня, решают программным путем.

6.6. Сеть ASI (Actuator Sensor Interface)

Основная задача этой сети - связать в единую информационную структуру устройства самого нижнего уровня автоматизируемого процесса (датчики и разнообразные исполнительные механизмы) с системой контроллеров.

ASI-интерфейс позволяет через свои коммуникационные линии передавать не только данные, но и запитывать датчики. Здесь используется принцип последовательной передачи на базовой частоте. Информационный сигнал модулируется на питающую частоту.

В качестве физической среды используется специальный незкранированный двухпроводный кабель с трапециевидным профилем. Этот кабель позволяет подключать датчики, устанавливаемые на подвижных частях механизмов. Топологией ASI-сети может быть шина, звезда, кольцо или дерево с циклом опроса 31 узла за 5 мс.

6.7. Сеть FOUNDATION FIELDBUS

Основная область применения этой сети - самый нижний уровень распределенной системы автоматизации с обвязкой устройств, работающих во взрывоопасных средах и использующих сеть как для информационного обмена, так и для собственной запитки.

Foundation Fieldbus (FF) — самый молодой и быстро растущий стандарт на промышленную сеть. FF представляет собой двухуровневый сетевой протокол, сочетающий черты мощной информационной магистрали для объединения компьютеров верхнего уровня и управляющей сети, объединяющей контроллеры, управляющие компьютеры, датчики и исполнительные механизмы.

На нижнем уровне в качестве физической среды передачи данных за основу взят стандарт IEC 61158-2, который позволяет использовать сеть FF на взрывоопасных производствах с возможностью запитки датчиков

непосредственно от канала связи. Скорость передачи информации на нижнем уровне составляет 31,5 Кбит/с.

На верхнем уровне) в настоящее время, как правило, используется FF HSE (High Speed Ethernet), основанный, как видно из названия, на сети Ethernet со скоростью 100 Мбит/с. Особенностью стандарта FF является то, что в нем определен дополнительный пользовательский уровень (User Layer), позволяющий, применяя предопределенные функциональные блоки, строить промышленные сети с распределенным интеллектом.

6.8. Сеть PROFIBUS

При построении многоуровневых систем автоматизации, как правило, стоят задачи организации информационного обмена между уровнями. В одном случае необходим обмен сообщениями на средних скоростях. В другом - быстрый обмен короткими сообщениями с использованием упрощенного протокола обмена (уровень датчиков). В третьем требуется работа в опасных участках производства (переработка газа, химическое производство). Для всех этих случаев PROFIBUS имеет решение. Сегодня под PROFIBUS понимается совокупность трех отдельных протоколов: PROFIBUS-FMS, PROFIBUS-DP и PROFIBUS-PA. Все три варианта протокола используют общий канальный уровень (уровень 2 OSI-модели).

Протокол PROFIBUS-FMS включает в себя дополнительные типы пакетов (Fieldbus Message Specification), появился первым и был предназначен для работы на так называемом цеховом уровне. Позволяет организовывать в одной сети работу нескольких активных станций.

Протокол PROFIBUS-DP был спроектирован для организации быстрого (до 12 Мбит/с) канала связи с датчиковым уровнем. Физическая среда передачи — экранированная витая пара стандарта RS-485. В основе алгоритма работы лежит модель циклического опроса каналов. Кроме этого, существует набор ациклических функций для конфигурирования, диагностики и поддержки сигналов.

Протокол PROFIBUS-PA — сетевой интерфейс, физическая среда передачи данных которого соответствует требованиям стандарта IEC 61158-2. Может применяться для построения сети, соединяющей исполнительные устройства, датчики и контроллеры, расположенные непосредственно во взрывоопасной зоне. Он может использоваться в качестве замены старой 4-20мА-технологии связи. Для коммутации устройств нужна всего одна витая пара, которая может одновременно использоваться и для информационного обмена и для запитывания устройств.

На одном физическом канале (RS485 или оптоволоконном) одновременно могут работать устройства PROFIBUS всех трех типов. Рабочая скорость передачи может быть выбрана в диапазоне 9,6-12000Ккбит/с.

PROFIBUS широко используется для модернизации и расширения возможностей существующих систем. Если требуется объединить в детерминированную сеть несколько контроллеров, оптимальным вариантом будет PROFIBUS-FMS. Для создания сети с централизованным интеллектом и распределенным вводом/выводом лучше всего подойдет PROFIBUS-DP.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Холмогоров В. Компьютерная сеть своими руками. Самоучитель.– СПб.: Питер, 2004. – 171с.

2. Электронная книга аппаратных средств. Hardware Book - http://www.hardwarebook.info/

3. Прохоров А. Интернет: как это работает.-СПб.: БХВ-Петербург, 2004. - 280 с.

4. Мамаев М. Телекоммуникационные технологии (Сети TCP/IP): учеб. пособие.– Владивосток : ВГУ ЭиС, 2001.

5. Олифер В.Г., Олифер Н.А. Компьютерные сети.-СПб.: Питер, 2003.- 863 с.

ОГЛАВЛЕНИЕ

1. Сети Ethernet 3
1.1. Классификация сетей Ethernet 3
1.2. Обозначение стандартов сетей Ethernet 4
1.3. Коаксиальный кабель 4
1.4. Витая пара 5
1.5. Правила обжима витой пары 5
1.6. Оптическое волокно 6
1.7. Топологии сетей 7
1.8. Концентратор Hub 11
1.9. Коммутатор Switch 11
1.10. Репитеры. 12
1.11. Настройка сети в Windows XP 12
1.12. Режимы передачи данных 17
2. Интерфейсы персонального компьютера 18
2.1. Последовательный интерфейс RS-232 18
2.2. Hyper Terminal 19
2.3. Параллельный порт LPT (порт принтера) 20
3. Internet 21
3.1. Доменные имена 21
3.2. DNS-сервер 23
3.3. Маршрутизация в сетях IP 27
3.4. Типы адресов (MAC, IP, DNS) 28
3.5. Протоколы ARP и RARP 30
4. Сетевые протоколы 32
4.1. Протокол межсетевого взаимодействия IP 37
4.2. Протокол DHCP 38
4.3. Протокол ICMP 41
4.4. Протокол EIGRP 43
4.5. Протокол RIP 45
4.6. Протокол OSPF 51
5. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ 54
5.1. Уровни модели OSI 56
5.2. Стек протоколов TCP/IP 64
6. ПРОМЫШЛЕННЫЕ СЕТИ 68
6.1. HART – протокол 68
6.2. Одноточечный режим передачи данных 69
6.3. Многоточечный режим передачи данных 70
6.4. Определение промышленной сети 72
6.5. Объем информационного сервиса 72
6.6. Сеть ASI (Actuator Sensor Interface) 73
6.7. Сеть FOUNDATION FIELDBUS 73
6.8. Сеть PROFIBUS 74
Библиографический список 76

Учебное издание

Дятлова Елена Павловна
Новиков Александр Игоревич

Структура и принцип работы вычислительных сетей АСУ

Учебное пособие

Редактор и корректор Т.А.Смирнова
Техн. редактор Л.Я. Титова
Темплан 2009 г., поз. 84

Подп. к печати 24. 09. 09. Формат 60 х 84 /16. Бумага тип.№ 1.
Печать офсетная. Уч.-изд. л.5,0. Усл.-печ. л.5,0. Тираж 100 экз.
Изд. № 84. Цена «С». Заказ 2448

Ризограф ГОУВПО Санкт-Петербургского государственного технологического университета растительных полимеров, 198095, СПб., ул. Ивана Черных, 4.