

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНЫХ СЕТЕЙ

Глецевич Иван Иванович
509-V

0

ВВЕДЕНИЕ

Версия 2.8

0.0.0.1

Цель учебной дисциплины: подготовить обучающегося к изучению специализированных дисциплин, связанных с компьютерными сетями.

Задача учебной дисциплины: освоение основных теоретических вопросов, связанных со структурной и функциональной организацией компьютерных сетей.

Базовыми для дисциплины «Теоретические основы компьютерных сетей» являются дисциплины «Конструирование программ и языки программирования» и «Схемотехника».

0.0.0.2

Разделы для изучения:

ВВЕДЕНИЕ

ИЕРАРХИЧЕСКИЕ МОДЕЛИ УПРАВЛЕНИЯ КОМПЬЮТЕРНЫМИ СЕТЯМИ

СОМ-ПОРТЫ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

ПАКЕТНАЯ ПЕРЕДАЧА ДАННЫХ

КАНАЛЬНОЕ КОДИРОВАНИЕ

ТОПОЛОГИИ КОМПЬЮТЕРНЫХ СЕТЕЙ

СЛУЧАЙНЫЕ МЕТОДЫ ДОСТУПА К МОНОКАНАЛУ

ДЕТЕРМИНИРОВАННЫЕ МЕТОДЫ ДОСТУПА К МОНОКАНАЛУ

АДРЕСАЦИЯ В КОМПЬЮТЕРНЫХ СЕТЯХ

МЕТОДЫ ВЗАИМОДЕЙСТВИЯ В ЗВЕНЕ СЕТИ ПЕРЕДАЧИ ДАННЫХ

ПРИКЛАДНЫЕ ЗАДАЧИ В КОМПЬЮТЕРНЫХ СЕТЯХ

СПЕЦИАЛИЗИРОВАННЫЕ КОМПЬЮТЕРНЫЕ СЕТИ ДЛЯ ПЕРЕДАЧИ ФАЙЛОВ И СООБЩЕНИЙ

СРЕДЫ ПЕРЕДАЧИ ДАННЫХ

СТРУКТУРИРОВАННЫЕ КАБЕЛЬНЫЕ СИСТЕМЫ

На экзамен будут вынесены все рассмотренные на лекциях разделы.

0.0.0.3а

Лабораторные работы:

1. Передача сообщений между двумя станциями посредством СОМ-портов (на базе эмулятора).
2. Пакетная передача и алгоритм бит- либо байт-стаффинга.
3. Код CRC либо Хэмминга.
4. Упрощенный вариант алгоритма CSMA/CD.
5. Упрощенный вариант алгоритма Token Ring.
6. Упрощенный вариант алгоритма TCP.
7. Обжим витой пары.
8. Отладочная команда tcpdump и программа Wireshark.

Для получения допуска к зачету необходимо выполнить и защитить все выданные лабораторные работы.

0.0.0.3b

Варианты выполнения лабораторных работ:

1. Написание программ.
2. Решение задач.
3. Составление алгоритмов.

Лабораторная база:

1. Операционная система Microsoft Windows.
2. Любой дистрибутив операционной системы Linux.
3. Среда разработки Microsoft Visual Studio.
4. Среда разработки gcc.
5. Программа Wireshark.
6. Программа Eterlogic Virtual Serial Ports Emulator (либо аналогичная).
7. Кабель UTP, разъемы RJ-45, инструмент для обжима витой пары.

0.0.0.4

Основная литература, дополняющая лекционный материал:

- [1] Таненбаум, Э. Компьютерные сети: 5-е изд. / Э. Таненбаум, Д. Уэзеропл. -- СПб. : Питер, 2018. -- 960 с.
- [2] Одом, Уэнделл Официальное руководство Cisco по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 100-105: Академическое издание / Уэнделл Одом. -- М. : Вильямс, 2017. -- 1088 с.
- [3] Теория прикладного кодирования : учеб. пособие, в 2 т. / под ред. проф. В. К. Конопелько. -- Минск : БГУИР, 2004. -- Т. 2. -- 398 с.
- [4] Семенов, А. Б. Структурированные кабельные системы : 5-е изд. / А. Б. Семенов, С. К. Стрижаков, И. Р. Сунчелей. -- М. : ЛАЙТ Лтд., 2011. -- 640 с.

0.0.0.5a

Основные стандарты и документация:

[5] PC16550D. Universal Asynchronous Receiver/Transmitter with FIFOs
[Электронный ресурс] : Datasheet / National Semiconductor Corp. --
Электронные данные. -- Режим доступа: PC16550D.pdf.

0.0.0.5b

[6] IEEE Standard for Ethernet. [Электронный ресурс] : IEEE Std 802.3-2015 / IEEE Computer Society. -- Электронные данные. -- Режим доступа: 802.3-2015.zip.

[7] IEEE Standard for Information technology. Telecommunications and information exchange between systems. Local and metropolitan area networks. Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications [Электронный ресурс] : IEEE Std 802.11-2016 / IEEE Computer Society. -- Электронные данные. -- Режим доступа: 802.11-2016.pdf.

[8] IEEE Standard for Information technology. Telecommunications and information exchange between systems. Local and metropolitan area networks. Specific requirements. Part 5: Token ring access method and physical layer specification [Электронный ресурс] : IEEE Std 802.5-1998 / IEEE Computer Society. -- Электронные данные. -- Режим доступа: 802.5-1998.pdf.

0.0.0.5c

[9] User Datagram Protocol [Электронный ресурс] : Request for Comments 768 / Internet Society. -- Электронные данные. -- Режим доступа: <https://tools.ietf.org/html/rfc768>.

[10] Transmission Control Protocol. DARPA Internet Program. Protocol Specification [Электронный ресурс] : Request for Comments 793 / Internet Society. -- Электронные данные. -- Режим доступа: <https://tools.ietf.org/html/rfc793>.

[11] TCP Congestion Control [Электронный ресурс] : Request for Comments 2581 / Internet Society. -- Электронные данные. -- Режим доступа: <https://tools.ietf.org/html/rfc2581>.

0.0.0.5d

[12] File Transfer Protocol (FTP) [Электронный ресурс] : Request for Comments 959 / Internet Society. -- Электронные данные. -- Режим доступа: <https://tools.ietf.org/html/rfc959>.

[13] Simple Mail Transfer Protocol [Электронный ресурс] : Request for Comments 5321 / Internet Society. -- Электронные данные. -- Режим доступа: <https://tools.ietf.org/html/rfc5321>.

[14] Post Office Protocol -- Version 3 [Электронный ресурс] : Request for Comments 1939 / Internet Society. -- Электронные данные. -- Режим доступа: <https://tools.ietf.org/html/rfc1939>.

[15] Internet Message Access Protocol -- Version 4rev1 [Электронный ресурс] : Request for Comments 3501 / Internet Society. -- Электронные данные. -- Режим доступа: <https://tools.ietf.org/html/rfc3501>.

[16] Hypertext Transfer Protocol -- HTTP/1.1 [Электронный ресурс] : Request for Comments 2616 / Internet Society. -- Электронные данные. -- Режим доступа: <https://tools.ietf.org/html/rfc2616>.

[17] Requirements for Internet Hosts -- Application and Support [Электронный ресурс] : Request for Comments 1123 / Internet Society. -- Электронные данные. -- Режим доступа: <https://tools.ietf.org/html/rfc1123>.

0.0.0.5е

[18] Commercial Building Telecommunications Cabling Standard [Электронный ресурс] : ANSI/TIA-568-C.1-2009 / Telecommunications Industry Association. -- Электронные данные. -- Режим доступа: TIA-586-C.1.pdf.

0.0.0.5f

[19] Wireshark User's Guide [Электронный ресурс]. -- Электронные данные.
-- Режим доступа: https://www.wireshark.org/docs/wsug_html_chunked/.

0.0.0.5g

- + Help and Support соответствующих ОС.
- + Литература, связанная с оборудованием Cisco.

0.0.0.6a

Основные категории технической документации:

1. Application Notes.
2. Configuration Guides (Manuals).
3. Command References.
4. Databooks.
5. Datasheets.
6. Hardware Installation (Getting Started) Guides.
7. Maintenance (Service) Guides.
8. Product Briefs.
9. Release Notes.
10. Specification Updates.
11. User's (Administrator's) Guides.
12. White Papers.

0.0.0.6b

**Cisco IOS
IP
Configuration Guide**
Release 12.2

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-7811741=
Text Part Number: 78-11741-02

Configuring Network Address Translation | **Configuring IP Addressing**

Figure 5 NAT Overloading Inside Global Addresses

The router performs the following process in overloading inside global addresses, as shown in Figure 5. Both host B and host C believe they are communicating with a single host at address 2.2.2.2. They are actually communicating with different hosts; the port number is the differentiator. In fact, many inside hosts could share the inside global IP address by using many port numbers.

1. The user at host 1.1.1.1 opens a connection to host B.
2. The first packet that the router receives from host 1.1.1.1 causes the router to check its NAT table:
 - If no translation entry exists, the router determines that address 1.1.1.1 must be translated, and sets up a translation of inside local address 1.1.1.1 to a legal global address.
 - If overloading is enabled, and another translation is active, the router reuses the global address from that translation and saves enough information to be able to translate back. This type of entry is called an *extended entry*.
3. The router replaces the inside local source address 1.1.1.1 with the selected global address and forwards the packet.
4. Host B receives the packet and responds to host 1.1.1.1 by using the inside global IP address 2.2.2.2.
5. When the router receives the packet with the inside global IP address, it performs a NAT table lookup, using the protocol, inside global address and port, and outside address and port as a key; translates the address to inside local address 1.1.1.1; and forwards the packet to host 1.1.1.1.

Host 1.1.1.1 receives the packet and continues the conversation. The router performs Steps 2 through 5 for each packet.

To configure overloading of inside global addresses, use the following commands in global configuration mode:

Step	Command	Purpose
1	<code>Router(config)# ip nat pool name start-ip end-ip [netmask netmask] prefix-length prefix-length</code>	Defines a pool of global addresses to be allocated as needed.
2	<code>Router(config)# access-list access-list-number permit source [source-wildcard]</code>	Defines a standard access list.

IPC-40 Cisco IOS IP Configuration Guide

0.0.0.7

Нумерация слайдов:

раздел.подраздел.пункт.подпункт
(и латинская буква, если слайд не помещается на экран)

Категории слайдов:

1. Ключевые теоретические (розовая рамка)
2. Теоретические
3. Дополнительные иллюстрации (серая рамка)
4. Примеры команд (салатовая рамка)
5. Темы для обсуждения (голубая рамка)

0.0.0.8

Обратите внимание на терминологию.

Термины по тексту приведены в наиболее часто используемом виде. Но при введении терминов указаны все корректные (по мнению автора) варианты, включая аббревиатуры, синонимы, переводы и транслитерацию.

Поскольку языковым терминологическим первоисточником в настоящее время является английский язык при использовании даже «устоявшихся» переводов для однозначности трактовки в скобках приведены оригинальные варианты на английском языке, как и принято.

Некоторые широко известные термины и названия использованы по тексту еще до их формального введения.

При классификациях, если это не противоречит оригинальным названиям, в отношении обобщенных понятий использован термин «тип», а конкретных -- «вид».

Слово «либо» означает строгую альтернативу -- исключающее или (xor).

0.0.0.9

Форматы структурных единиц передаваемой по сети информации отображены в виде рисунков (в линию, в виде матрицы по байтам) и регулярных выражений (BNF) -- в зависимости от сложности.

Младшие биты на рисунках расположены справа.

По умолчанию использована десятичная система счисления.

Взаимодействия нескольких объектов во времени описаны в виде алгоритмов (текст по шагам, схема программы) и рисунков (диаграмма взаимодействия, стрелки с цифрами между объектами) -- в зависимости от ситуации.

Без видео, для улучшения обзора.

На слайдах использованы условные графические обозначения.

Опять же, обратите внимание на стили.

0.0.0.10

Курсивом выделены ключевые термины, находящиеся в широком обиходе.

Специфические термины, относящиеся к конкретным технологиям и используемые более узко, не выделены.

Красным цветом (шрифта) выделены исправленные обнаруженные ошибки, а также внесенные изменения и дополнения -- в сравнении с предыдущими версиями.

0.0.0.11

Примеры, показанные на разных слайдах, по умолчанию не зависят друг от друга.

0.0.1.1

Попробуйте дать определение компьютерной сети.

0.0.1.2

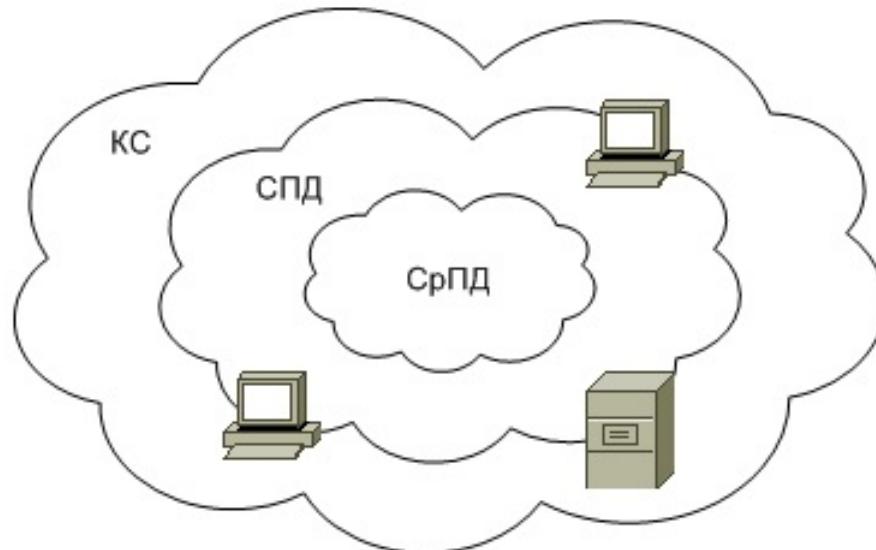
Под *компьютерной сетью* (КС) понимают совокупность различных технических средств (то есть самих компьютеров и другого оборудования), предназначенная для передачи компьютерной информации (то есть файлов и сообщений) на относительно большие расстояния (то есть за пределы компьютеров).

0.0.1.3

Любую КС можно рассматривать с двух точек зрения:

1. Программной.
2. Аппаратной.

0.0.1.4



В основе любой КС лежит так называемая *сеть передачи данных* (СПД) - - Data Communication Network (DCN), которая может задействовать различные *среды передачи данных* (СрПД -- аббревиатура нестандартная) (media).

Иногда в составе СПД выделяют базовую (опорную) СПД.

0.0.1.5

Все устройства в составе СПД можно разделить на две четко разделяющиеся группы:

1. Оконечные (end devices) -- находятся по периметру СПД.
2. Посредники (intermediary devices) -- составляют ядро СПД.

0.0.1.6

Приведите примеры окончных устройств.

Приведите примеры устройств-посредников.

0.0.1.7

Условные графические обозначения ряда сетевых устройств и сред (в нотации Cisco).



-- стационарная пользовательская станция



-- мобильная пользовательская станция



-- сервер



-- маршрутизатор



-- коммутатор



-- СрПД LAN



-- СрПД WAN

Условные графические обозначения других сетевых устройств и сред будут введены постепенно по мере надобности.

0.0.1.8

Весь трафик в СПД традиционно разделяют на три базовых типа:

1. *Обычные компьютерные данные (data).*
2. *Голос (voice).*
3. *Видео (video).*

Каждый тип обладает характерными особенностями.

СПД, поддерживающие пересылку разнородного трафика, в нотации Cisco называют *конвергированными* (converged networks).

0.0.1.9

Особенности трафика обеспечиваются так называемым *качеством обслуживания* -- Quality of Service (QoS).

Традиционные виды компьютерных данных без исключения, по умолчанию, обслуживаются по принципу: «Все делается для доставки пакетов, но при этом ничего не гарантируется» (best efforts), что, по сути, является отсутствием QoS.

Гарантии «возникают» при работе с голосом и видео.

0.0.1.10

В рамках предоставляемой оборудованием СПД *полосы пропускания* (bandwidth) можно выделить реально задействованную ее часть (throughput) и полезную составляющую этой задействованной части (goodput) -- без учета служебного трафика.

0.0.2.1a

С одной стороны, выделяют:

1. Local Area Networks (LANs) -- локальные КС (ЛКС).
2. Wide Area Networks (WANs) -- глобальные КС (ГКС).
- +3. Metropolitan Area Networks (MANs) -- городские КС (устоявшейся русскоязычной аббревиатуры нет).
- +4. Personal Area Networks (PANs) -- личные КС (устоявшейся аббревиатуры нет).
- +5. Remote Access Services (RASes) -- КС для подключения удаленных пользователей (teleworkers) (так же устоявшейся аббревиатуры нет).
- +6. Data Center (Centre) Networks -- КС центров обработки данных.
- +7. Home Networks -- домашние КС.
- +8. Industrial Networks -- промышленные КС.

С другой:

1. Intranets -- внутренние КС предприятий и организаций.
2. Internets -- КС публичного доступа.

0.0.2.1b

LAN выделяют прежде всего территориально -- в современном понимании, охватывает территорию не более кампуса, но при этом подразумевают определенные технологии.

Intranet обычно выделяют по ведомственной принадлежности пользователей.

WAN выделяют прежде всего технологически и, в общем случае, может охватывать произвольную территорию .

Практически все Internets сейчас интегрированы в одну сборную одноименную сеть.

MAN представляет собой промежуточный вариант между LAN и WAN.

PAN позволяет подключить к компьютеру периферийные устройства.

RAS существует в контексте WAN.

Home, Datacenter и Industrial Networks являются специализированными вариантами LAN.

Intranet почти всегда имеет связь с Internet.

0.0.2.2

Перечислите особенности, например, промышленных сетей.

0.0.2.3

Кроме того, сети могут быть:

1. *Изолированными (isolated).*
2. *Открытыми для прослушивания (open).*

0.0.2.4a

С точки зрения организации взаимодействия КС могут быть:

1. Сильносвязанными.
2. Слабосвязанными.

0.0.2.4b

В случае сильносвязанной КС подразумеваю наличие так называемой хост-ЭВМ (*host*) с одной стороны и *терминала* (*terminal*) -- с другой.

Хост является основным вычислительным компонентом.

Под терминалами подразумеваю исключительно устройства для ввода и отображения информации, следовательно, они без хоста бесполезны.

Совокупность хоста и подключенных к нему терминалов принято называть *рабочей станцией* (*workstation*).

Терминал администратора, обычно подключаемый особым образом, называют *консолью* (*console*).

Мы имеем дело с *хост-терминальной моделью*.

0.0.2.4c

В случае слабосвязанной КС подразумеваю наличие *сервера (server)* с одной стороны и *клиента (client)* -- с другой.

Клиентские ЭВМ, обслуживающие запросы пользователей, являются активными компонентами.

Сервер либо серверы, являющиеся пассивными компонентами, в свою очередь, обслуживаются запросы клиентов.

Как клиенты, так и серверы могут работать независимо, связываясь по мере необходимости.

Мы имеем дело с *клиент-серверной моделью*.

0.0.3.1

С точки зрения общей организации работы сетевых устройств, в первую очередь касательно WANs и RASes, принято выделять два типа оборудования:

1. Оконечное оборудование данных (ООД) -- Data Terminal Equipment (DTE).
2. Аппаратура передачи данных (АПД) -- Data Communication Equipment или, по-другому, Data Circuit-terminating Equipment (DCE).

Термины происходят из традиционной телефонии.

ООД находится на самой границе СПД и «концентрирует», то есть создает и потребляет передаваемые информационные потоки.

АПД находится в пределах СПД и «транслирует», то есть позволяет передавать и принимать информационные потоки.

Разница заключается и в синхронизации передаваемых потоков. Первосточником синхронизации обычно является АПД.

Понятие ООД хорошо соотносится с понятием оконечных устройств, а понятие АПД хорошо соотносится с понятием устройств-посредников. Но, поскольку, термины ООД и АПД связаны с определенными технологиями, а задействующие эти технологии устройства часто встречаются в самых разных частях СПД, знаки равенства ставить некорректно.

0.0.4.1

Какие вы знаете сетевые стандарты?

0.0.4.2

Все стандарты, в том числе в области КС, делят на:

1. Международные (например, ISO/IEC).
2. Европейские (например, EN).
3. Американские (например, ANSI/TIA/EIA).

Стандарты лишь формализуют определенные требования в той или иной предметной области.

Стандарты могут носить предварительный (preliminary) или временный (interim) характер. Могут включать дополнения (annexes, addendums = addenda) и списки обнаруженных ошибок (errata). Могут устаревать или замещаться другими стандартами (obsolete).

Практическим (или теоретическим) воплощением стандарта является так называемая реализация (implementation).

Сертификация (certification) позволяет определить факт соответствия стандарту.

0.0.4.3

В 1980 г. при IEEE был создан специальный комитет по стандартизации КС, результатом работы которого стало множество стандартов 802.x.

Сейчас наибольший интерес представляют:

1. 802.3 -- Ethernet.
2. 802.11 -- Wi-Fi.
3. 802.16 -- WiMax.

0.0.4.4

Стандарты Ethernet по пропускной способности делят на три группы:

1. Ethernet -- до 10 Mbit/s включительно.
2. Fast Ethernet -- 100 Mbit/s.
3. Gigabit Ethernet -- 1, 10, 100, 40, 25 Gbit/s и Multigigabit.

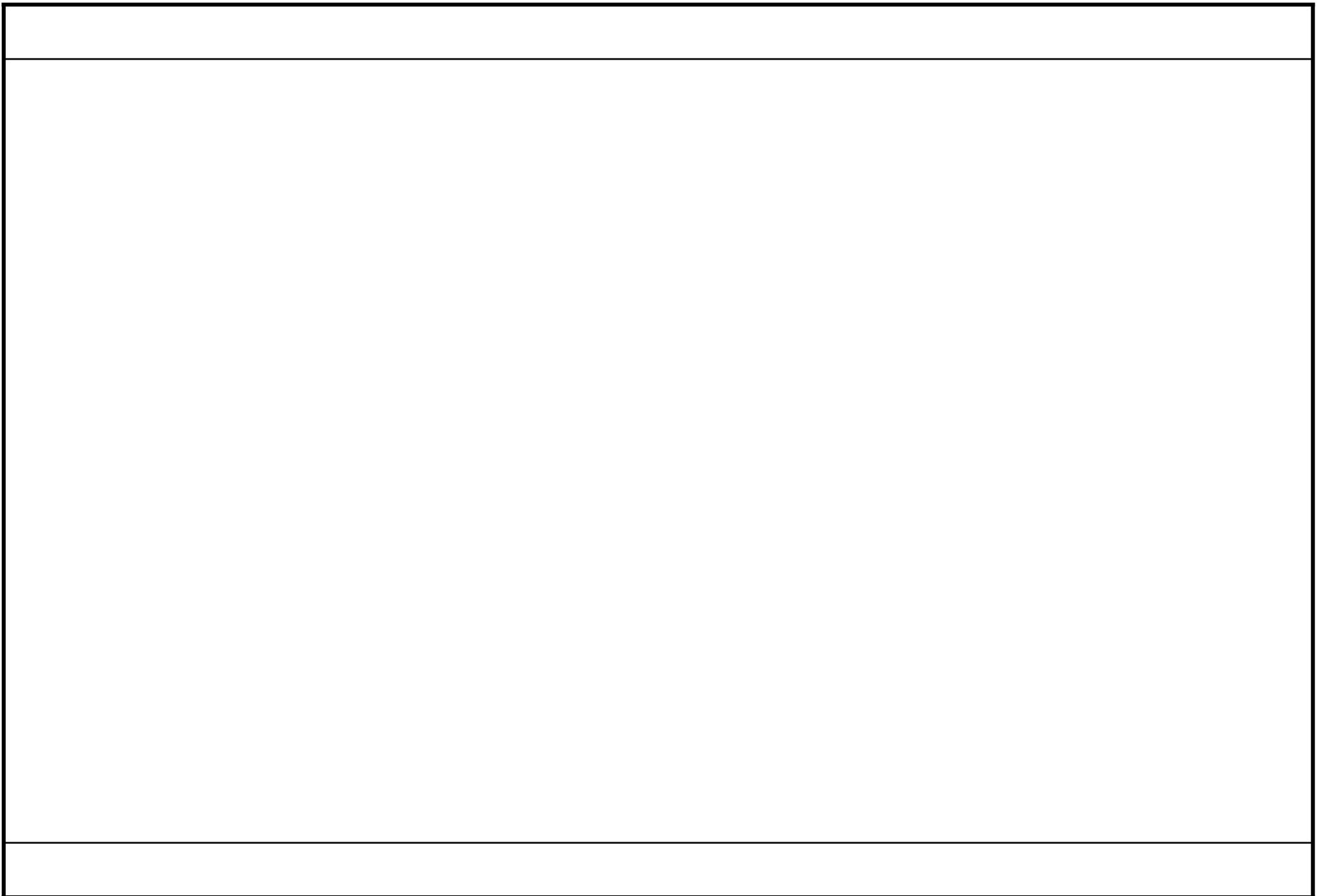
0.0.5.1

Повсеместное внедрение ЛКС привело к необходимости их интеграции в инфраструктуру зданий и сооружений.

Структурированная кабельная система (СКС) -- Structured Cabling System (SCS) -- представляет собой упорядоченную гетерогенную коммуникационную подсистему зданий и сооружений.

Выделяют следующие стадии работ, связанных с СКС, для каждой из которых предусмотрен собственный набор стандартов:

1. Проектирование.
2. Монтаж.
3. Эксплуатация.



ИЕРАРХИЧЕСКИЕ МОДЕЛИ УПРАВЛЕНИЯ КОМПЬЮТЕРНЫМИ СЕТЯМИ

1.0.1.1

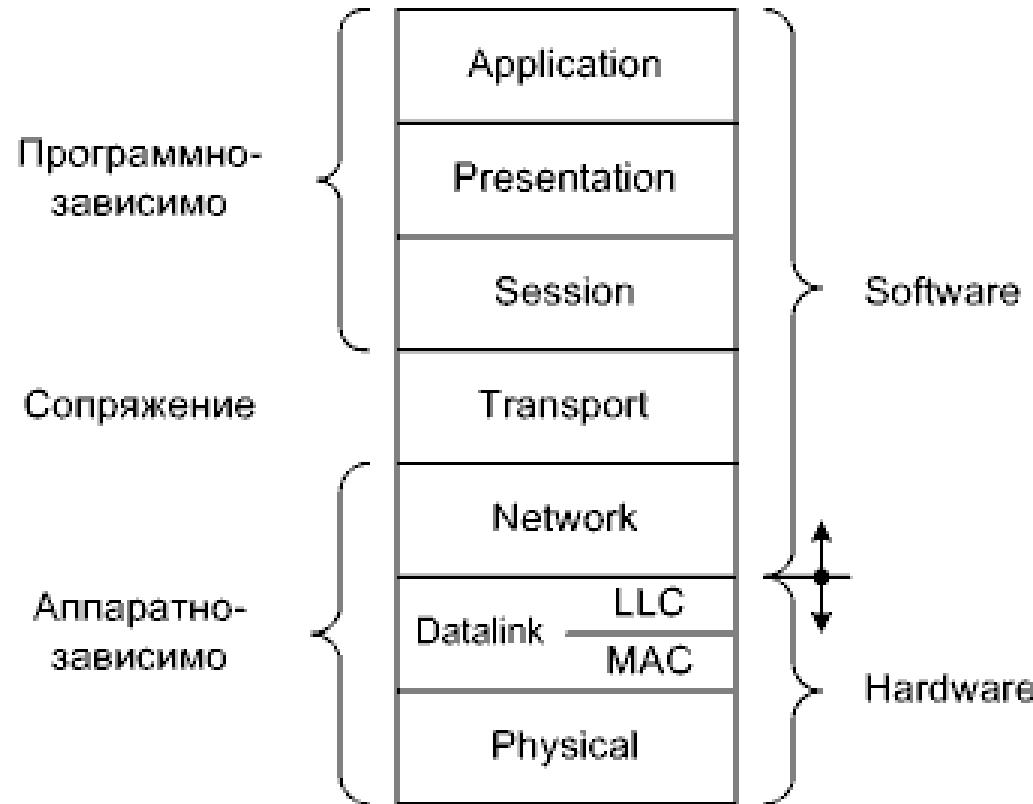
Из всех моделей КС наиболее фундаментальной является *открытая модель взаимодействия систем* -- Open System Interconnection (OSI), разработанная ISO.

1.0.1.2

Что понимают под открытостью модели?

1.0.1.3

Модель включает семь уровней.



На вершине иерархии находится человек, но абонентами КС являются взаимодействующие программы.

1.0.2.1

Как вы думаете, зачем нужен физический уровень?

И какую область очерчивает физический уровень?

1.0.2.2

На *физическом* (physical) уровне формализуют подключение того либо иного сетевого устройства к СрПД.

Соответственно в пространстве физический уровень охватывает «точку» подключения.

1.0.2.3

Какие задачи возложены на физический уровень?

Попробуйте перечислить специфические понятия физического уровня?

1.0.2.4

Специфическими понятиями физического уровня являются:

- среда;
- разъем (физический порт);
- несущая (частота);
- модуляция;
- сигнал.

Фундаментальная задача физического уровня заключается в передаче сигнала.

1.0.2.5

Что такое несущая (carrier)?

Что такое модуляция (modulation)?

Назовите базовые способы модуляции.

1.0.2.6

Чем отличается симметричная электрическая цепь от несимметричной?

1.0.2.7

Подумайте, зачем нужен канальный уровень?
Какую область очерчивает канальный уровень?
Какие новые задачи возникают при переходе от физического уровня к
канальному?

1.0.2.8

На *канальном* (datalink) уровне формализуют взаимодействие станций в пределах сегмента.

Любое устройство, способное передавать или принимать сетевой трафик принято называть *станцией* или, по-другому, *узлом* (node). Примерами станций могут быть: ПК, сервер, маршрутизатор и так далее.

Физически любая КС состоит из некоторого, большего или меньшего, количества сегментов. *Сегментом* (segment) называют множество станций, объединенных посредством одной СрПД, то есть «видящих» друг друга непосредственно. Технологически сегменты могут быть самыми разными.

В традиционном понимании СрПД соответствует *физическому соединению* (link). Но многие современные технологии предполагают опциональное или обязательное наличие в СрПД «прозрачных» устройств-посредников, таких как преобразователи сред или коммутаторы.

1.0.2.9

Специфическими понятиями канального уровня являются:

- сегмент сети;
- физическая и логическая топология сегмента;
- пакет (кадр);
- бит- и байт-стаффинг;
- адресация в пределах сегмента;
- канальный код;
- код проверки целостности пакета (кадра);
- алгоритм доступа к моноканалу.

Эти понятия более подробно будут рассмотрены далее в соответствующих разделах.

1.0.2.10

Каждый из уровней модели OSI может быть реализован достаточно сложно, но канальный уровень особенно сложен. Поэтому его разделяют на два подуровня:

1. MAC (Media Access Control) -- контроль доступа к СрПД.
2. LLC (Logical Link Control) -- контроль логического соединения.

На подуровне MAC, более низком, выполняется взаимодействие с физическим уровнем, то есть средозависимые операции, такие как формирование и распознавание пакетов, адресация, канальное кодирование и другие.

На подуровне LLC, более высоком, выполняется взаимодействие с сетевым уровнем, то есть средонезависимые операции, такие как разбиение данных на пакеты, сборка данных из пакетов, определение соответствующей подсистемы сетевого уровня и другие.

1.0.2.11

Подумайте, зачем нужен сетевой уровень?

Какую область очерчивает сетевой уровень?

1.0.2.12

Сетевой уровень позволяет «выйти» за пределы сегмента.

На *сетевом* (network) уровне формализуют построение полноценной КС произвольного масштаба, охватывающей произвольное количество сегментов.

1.0.2.13

Специфическими понятиями сетевого уровня являются:

- пакет (собственно пакет);
- адресация в пределах всей КС;
- маршрутизация.

1.0.2.14

Подумайте, зачем нужен транспортный уровень?
Какую область очерчивает транспортный уровень?

1.0.2.15

Транспортный уровень позволяет перейти от оборудования к программам.

На *транспортном* (*transport*) уровне формализуют использование программным обеспечением сетевого оборудования, то есть как отдельно взятым программам предоставляется «транспорт».

1.0.2.16

Специфическими понятиями транспортного уровня являются:

- пакет (сегмент сообщения);
- программный порт;
- логическое соединение;
- надежность доставки;
- алгоритм борьбы с заторами в СПД.

1.0.2.17

Подумайте, зачем нужен сеансовый уровень?

1.0.2.18

Сеансовый или сессионный (session) уровень позволяет предоставить доступ к транспорту всем программам в многозадачном окружении.

1.0.2.19

Кроме собственно сессии, имеются еще два основных специфических понятия сеансового уровня:

- программный порт;
- алгоритм мультиплексирования программ.

В практических реализациях сеансовый уровень выражен слабо и обычно его **совмещают** с транспортным.

1.0.2.20

Подумайте, зачем нужен прикладной уровень?

1.0.2.21

Прикладной (application) уровень призван решать конкретные пользовательские задачи с помощью КС.

1.0.2.22

Приведите примеры прикладных задач?

1.0.2.23

Примерами прикладных задач могут служить:

- пересылка файлов между компьютерами;
 - пересылка электронных писем;
 - поддержка удаленных текстовых и графических терминалов, в том числе для администрирования;
 - пересылка мультимедийных документов;
 - обмен «мгновенными» сообщениями;
 - совместная разработка чего-либо;
- и другие.

Плюс, выделяемые особо, как не свойственные традиционным компьютерным сетям, задачи пересылки голоса и видео в реальном времени. При этом, качество обслуживания «возникает» и на всех нижестоящих уровнях.

Специфических понятий прикладного уровня великое множество и они зависят от решаемых задач.

1.0.2.24

Наконец, подумайте, зачем нужен уровень представления?

1.0.2.25

Уровень *представления* (presentation) позволяет адаптировать прикладную информацию в форму, приемлемую для передачи по КС, то есть является прослойкой между программами и транспортом.

1.0.2.26

Назовите основные задачи, решаемые на уровне представления (их две)?

1.0.2.27

Основными задачами уровня представления являются:

- кодирование информации (включая возможное сжатие) с целью обеспечения ее правильной интерпретации в последующем;
- шифрование информации с целью обеспечения ее защиты при пересылке по открытым для прослушивания сетям.

Поскольку обычно уровень представления «привязан» к прикладному уровню, в реализациях эти уровни часто **совмещают**.

1.0.2.28

Взаимодействие в рамках модели OSI может быть «вертикальным» и «горизонтальным»:

1. *Интерфейс* (interface) -- это правила взаимодействия между пространственно совмещенными соседними уровнями модели OSI.
2. *Протокол* (protocol) -- правила взаимодействия между пространственно разнесенными одинаковыми уровнями модели OSI.

И в том, и в другом случае предполагают определенную абстракцию.

1.0.3.1

Исторически сложились два основных семейства протоколов:

1. TCP/IP.
2. IPX/SPX.

В настоящее время TCP/IP полностью доминирует. IPX/SPX почти не используют, но вкратце будет рассмотрен позже.

1.0.3.2

Application	FTP	Telnet	SMTP	DNS	HTTP	...
Presentation						
Session	TCP			UDP		
Transport						
Network	ICMP	RIP	OSPF	...		
			IP			
	ARP			RARP		
Datalink						
Physical	Ethernet	Token Ring		FR	...	

Описания протоколов будут вводиться в дальнейшем по мере надобности.

Семейство протоколов TCP/IP

1.0.3.3

Семейство протоколов TCP/IP описано в стандартах RFC (Request For Comments).

1.0.3.4

[Docs] [txt|pdf] [draft-ietf-ipngwg...] [Diff1] [Diff2] [Errata]

Updated by: [5095](#), [5722](#), [5871](#), [6437](#), [6564](#), [6935](#),
[6946](#), [7045](#), [7112](#)

DRAFT STANDARD

[Errata Exist](#)

Network Working Group
Request for Comments: 2460
Obsoletes: [1883](#)
Category: Standards Track

S. Deering

Cisco

R. Hinden

Nokia

December 1998

Internet Protocol, Version 6 (IPv6) Specification

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

This document specifies version 6 of the Internet Protocol (IPv6), also sometimes referred to as IP Next Generation or IPng.

Table of Contents

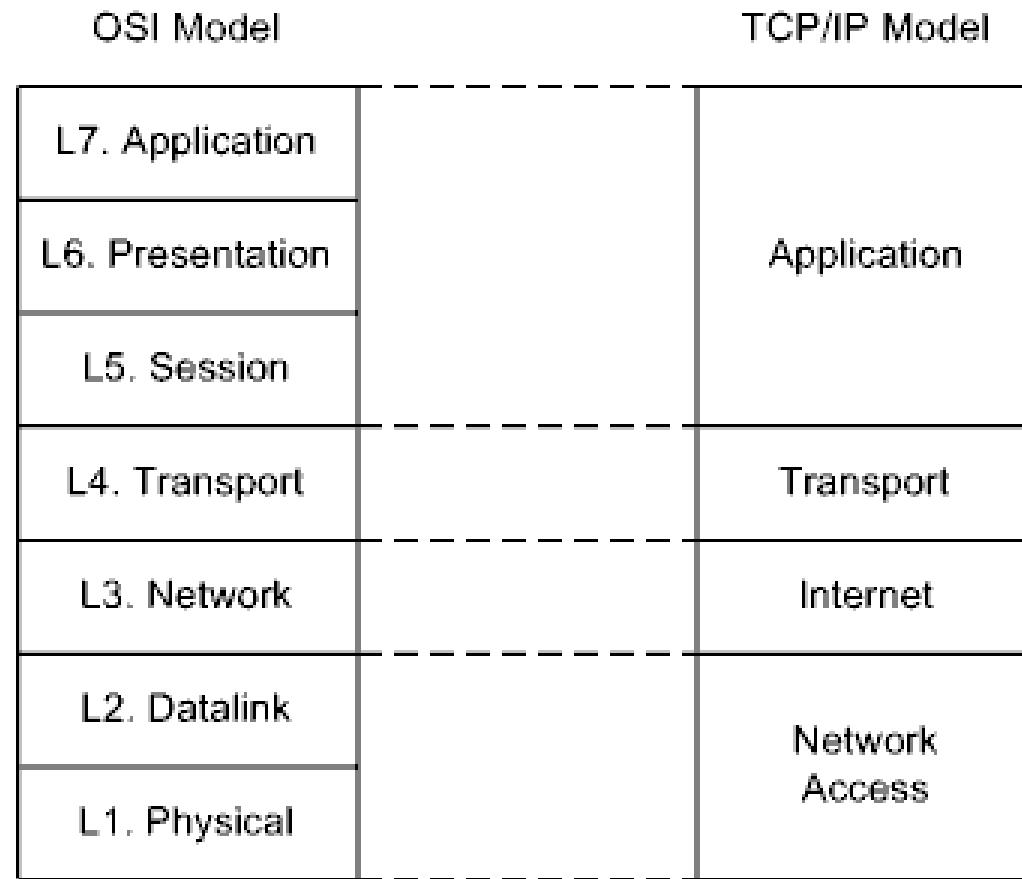
1. Introduction	2
2. Terminology	3
3. IPv6 Header Format	4
4. IPv6 Extension Headers	6
4.1 Extension Header Order	7
4.2 Options	9
4.3 Hop-by-Hop Options Header	11
4.4 Routing Header	12

Пример RFC

1.0.3.5

С семейством протоколов TCP/IP связана одноименная модель.

1.0.3.6



Сопоставление модели TCP/IP с моделью OSI

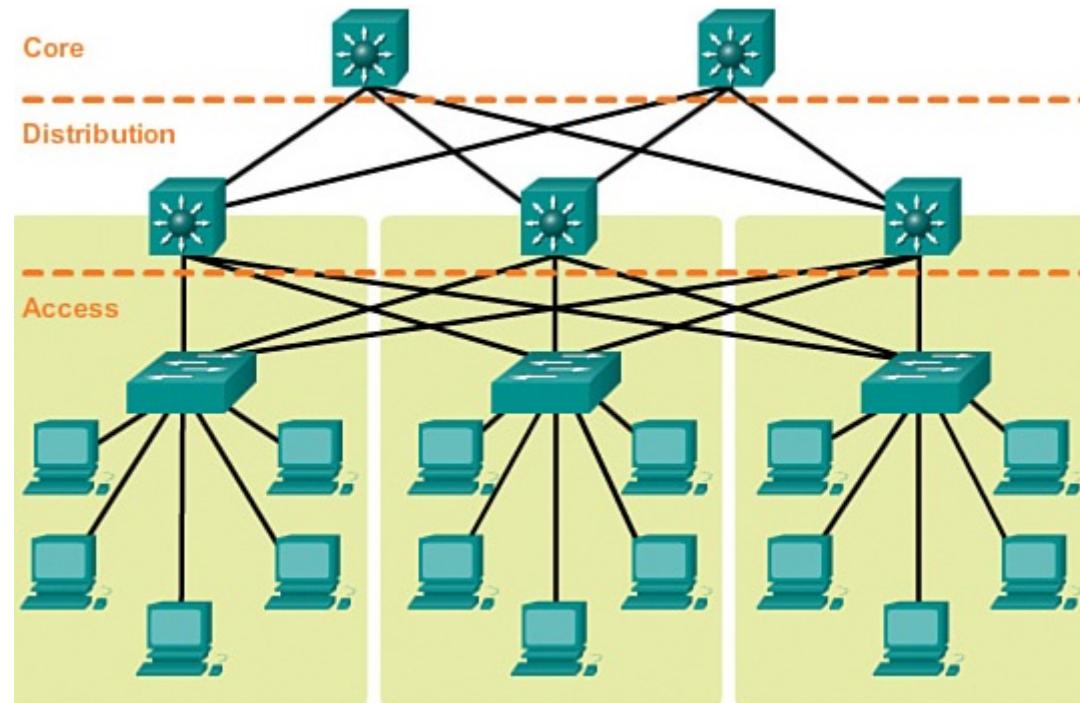
1.0.4.1

Компания Cisco на основе многолетнего опыта проектирования сетей разработала собственную иерархическую сетевую модель (Cisco hierarchical network model), которую рекомендует использовать в корпоративных (enterprise) сетях разного масштаба.

1.0.4.2a

Модель включает три уровня:

1. Access -- доступ.
2. Distribution (иногда aggregation) -- распределение.
3. Core -- ядро.



1.0.4.2b

Уровень доступа предназначен для обеспечения подключений к КС оконечных пользователей. Особое внимание здесь уделяют предоставлению пользователям требующихся им ресурсов.

Уровень распределения предназначен для обеспечения взаимодействия в пределах групп пользователей. Особое внимание здесь уделяют резервированию соединений.

Уровень ядра предназначен для обеспечения высокоскоростной связи между относительно удаленными группами пользователей. Особое внимание здесь уделяют характеристикам трафика.

На всех уровнях значительное место отведено разграничению трафика с целями защиты пользователей друг от друга и защиты КС от пользователей.

При этом всем, технологии могут быть различными. Догма нет. Привязка конкретной технологии к тому или иному уровню требует ее понимания. Технологии Cisco будут рассматриваться в дальнейшем.

1.0.5.1

При разговоре о структурной и функциональной организации КС неизбежно возникает вопрос о *сетевой архитектуре* (network architecture).

В классическом представлении под архитектурой (в том числе сетевой) понимают «проекцию» вычислительной структуры на пользователя, то есть как пользователь «видит» оборудование.

1.0.6.1

Примеры сетевых архитектур, которые активно пропагандирует Cisco:

1. Cisco SecureX.
2. Cisco Borderless Network (в рамках BYOD).
3. Cisco Collaboration.
4. Cisco Unified Data Center.

Конечно, архитектуры от Cisco ориентированы в первую очередь на собственные аппаратные и программные решения.

Просматривается, что сейчас все больший уклон делают в сторону защищенного подключения мобильных пользователей, виртуализации и облачных вычислений. Такие архитектуры позволяют строить так называемые КС с нечетко очерченной границей (borderless).

1.0.6.2

Архитектура Cisco SecureX включает пять основных компонентов:

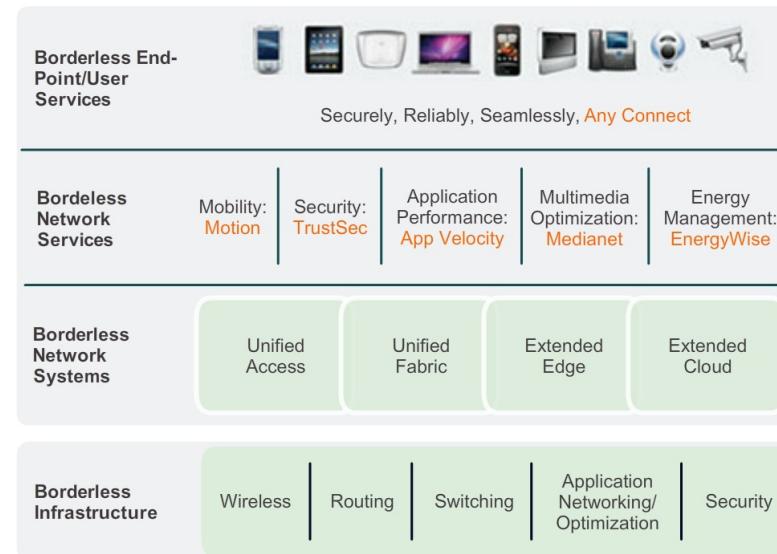
1. Scanning Engines -- движки для отслеживания различных угроз.
2. Delivery Mechanisms -- механизмы «внедрения» сканирующих движков.
3. Security Intelligence Operations -- операции для изоляции вредоносного трафика.
4. Policy Management Consoles -- консоли для централизованного управления политикой безопасности.
5. The Next-generation Endpoint -- самые современные оконечные устройства.



1.0.6.3

Архитектура Cisco Borderless Network -- позволяет создать среду BYOD (Bring Your Own Device) и включает четыре основных компонента:

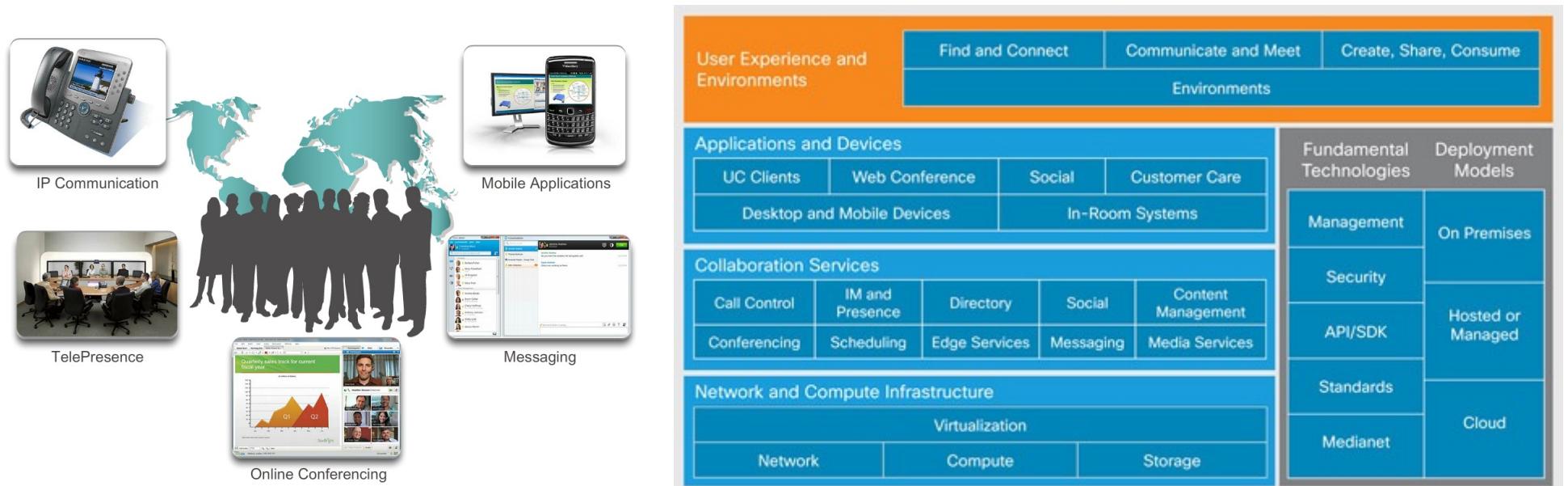
1. Cisco Borderless End Point/User Services -- сервисы для подключения самых разнообразных пользовательских устройств.
2. Cisco Borderless Network Services -- сервисы оптимизации взаимодействия между устройствами.
3. Cisco Borderless Network Systems -- подсистемы для организации взаимодействия между пользователями и организации взаимодействия пользователей с централизованными ресурсами.
4. Cisco Borderless Infrastructure -- инфраструктура, включающая программное и аппаратное обеспечение для реализации требующихся сервисов и подсистем.



1.0.6.4

Архитектура Cisco Collaboration включает четыре основных компонента:

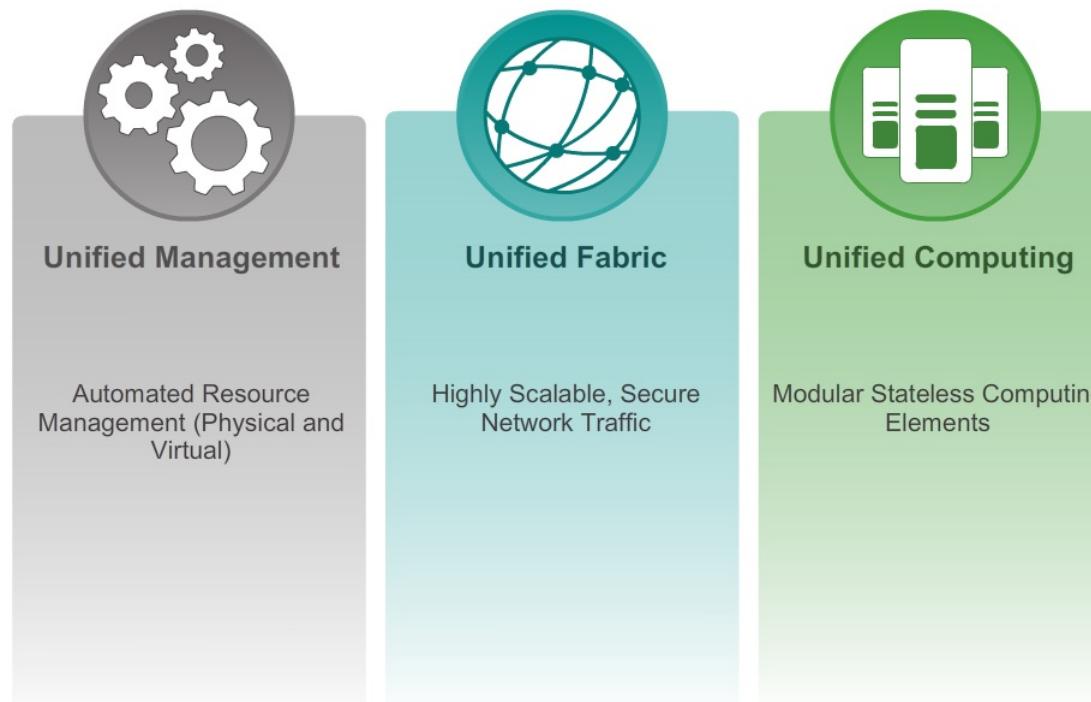
1. Cisco TelePresence -- современное техническое средство для организации телеконференций.
2. Collaboration Applications -- приложения для совместной работы, позволяющие участвовать в телеконференциях и создавать мультимедийный контент для них.
3. Customer Collaboration -- набор программных и аппаратных средств для организации взаимодействия между производителями и потребителями услуг.
4. Cisco Unified Communications -- набор средств для контроля, управления и оптимизации всех коммуникаций с «одной точки».

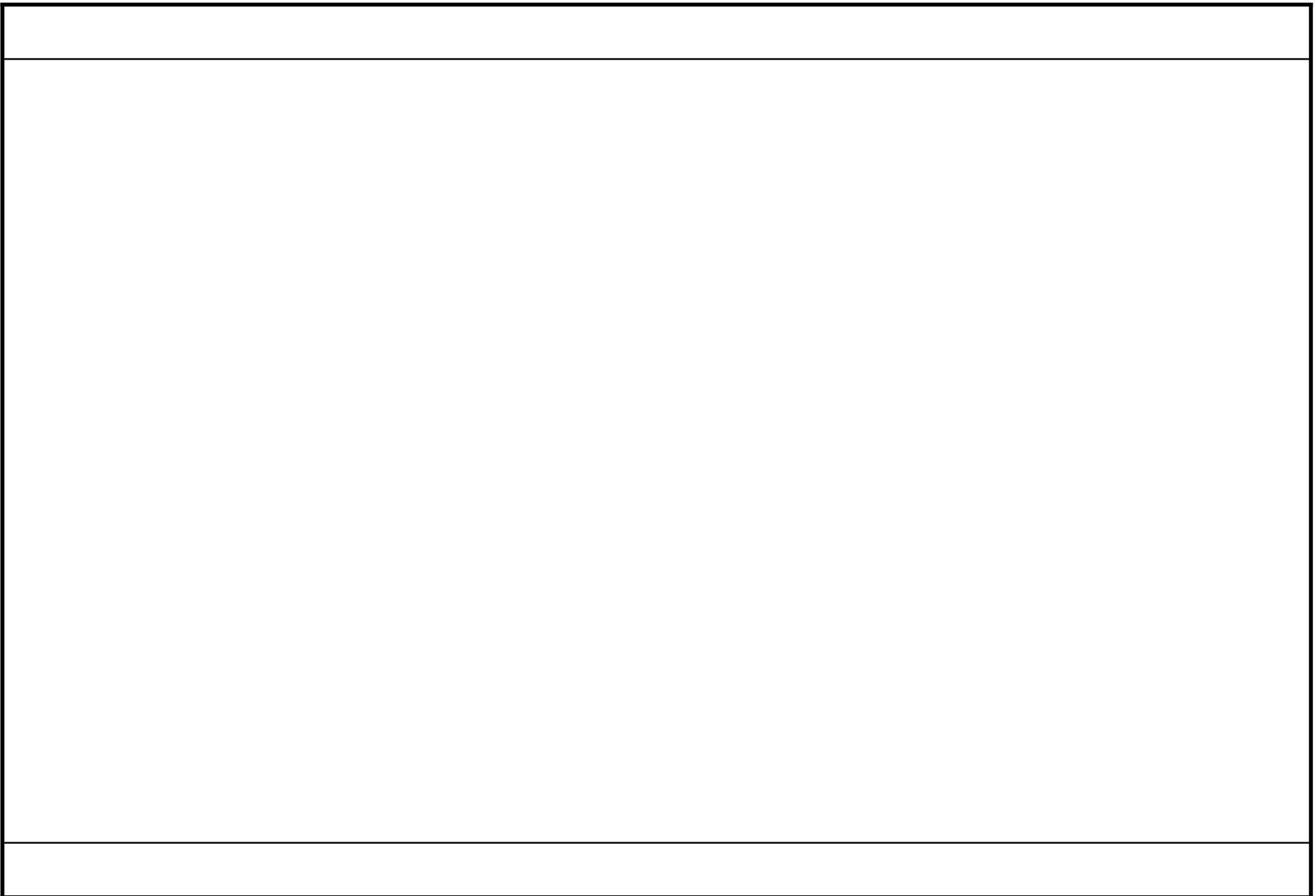


1.0.6.5

Архитектура Cisco Unified Data Center включает три основных компонента:

1. Cisco Unified Management -- автоматизированное управление ресурсами (физическое и виртуальное).
2. Cisco Unified Fabric -- защищенный сетевой трафик произвольного объема.
3. Cisco Unified Computing -- модульные динамически формируемые вычислительные элементы.





СОМ-ПОРТЫ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

2.0.1.1

История развития последовательного (serial) или, по-другому, коммуникационного (СОМmunication) порта неразрывно связана с развитием элементной базы.

Применительно к ПК разработчиком как базовой архитектуры так и типовых схем оборудования являлась и до сих пор является компания Intel.

2.0.1.2a

В развитии СОМ-порта ПК можно выделить следующие основные этапы (следует отметить, что этот процесс сильно коррелирует с развитием СОМ-портов всех типов компьютерных систем):

1. В свое время (семидесятые годы XX века), в составе периферийной части комплекта микросхем поддержки микропроцессора 8080, компания Intel разработала два контроллера последовательного порта.

Один из них, 8250, получил название UART (Universal Asynchronous Receiver/Transmitter) -- универсальный асинхронный приемник-передатчик.

Второй, 8251, получил название USART (Universal Synchronous/Asynchronous Receiver/Transmitter) -- универсальный синхронно-асинхронный приемник-передатчик.



2.0.1.2b

Эти контроллеры были рассчитаны на подключение по шине X-Bus (шина ввода-вывода, внутрисхемный восьмибитный предшественник системной шины ISA) и поэтому без труда были перенесены в первые ПК на базе процессора 8086 и его модификаций (то есть компьютеры класса IBM PC XT) с тогда наиболее распространенной системной шиной ISA.

Совместно с контроллером параллельного порта 8255, микросхема UART либо USART устанавливалась на плату специального адаптера и подключались к материнской плате ПК посредством разъема системной шины.

В это же время возникла традиция устанавливать последовательные порты парами (COM1 и COM2).

2.0.1.3а

2. Времена доминирования процессоров 80286 -- Intel486 (то есть компьютеры класса IBM PC AT и IBM PS/2) ознаменованы постепенно набравшими силу интеграционными процессами.

На первом этапе происходило распространение и развитие самих контроллеров.

В СССР был создан аналог 8251 под названием KP580BB51A, который и стал массово применяться в серии ЕС ПК.

На Западе же, наоборот, развитие получила микросхема 8250.

2.0.1.3b

Апофеозом достаточно быстрого усовершенствования 8250 стали несколько UART, среди которых следует выделить 16550, причем это была разработка уже не Intel, а National Semiconductor. Именно эта микросхема стала де facto стандартной на длительное время (архитектурная совместимость сохраняется вплоть до настоящего времени).

16550 имеет два основных преимущества перед 8250:

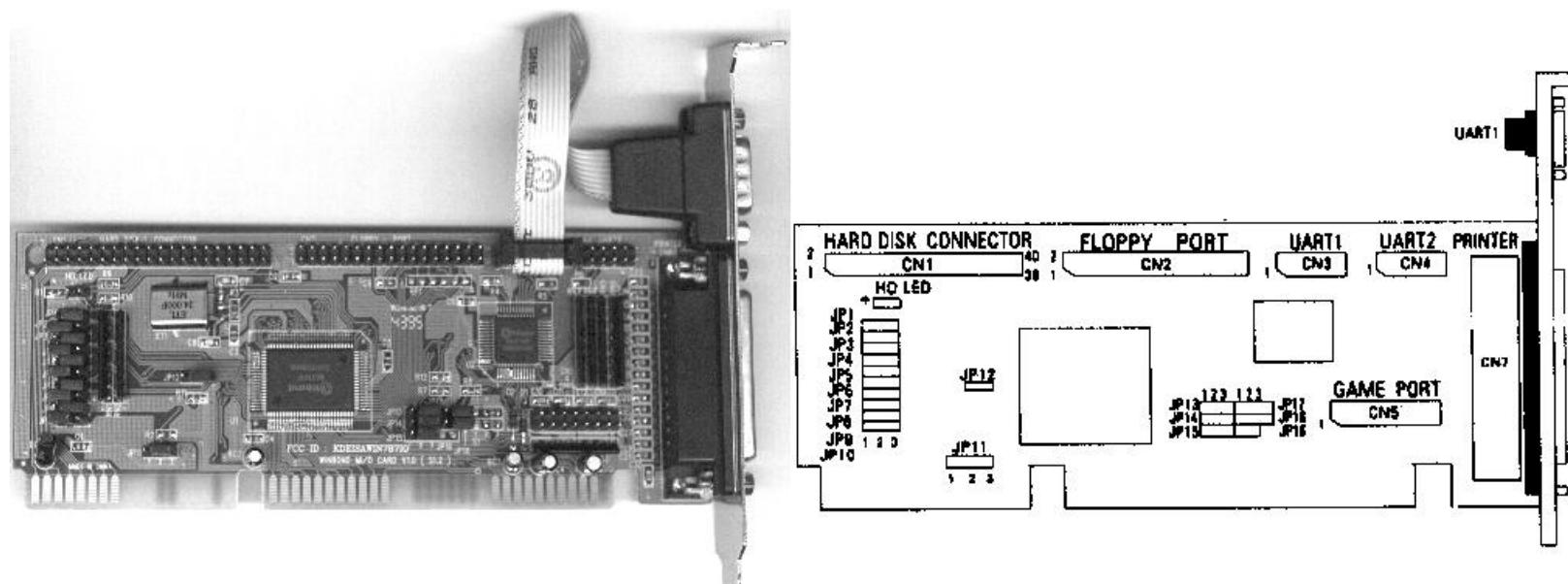
- более высокая пропускная способность последовательного интерфейса (максимальная стандартная пропускная способность увеличена с 9600 baud до 115200 baud);
- возможность буферизации (две очереди FIFO по 16 байт -- на стороне передатчика и на стороне приемника).



2.0.1.4

В дальнейшем интеграционные процессы привели к появлению так называемых мультикарт -- подключаемых посредством разъема системной шины (по-прежнему обычно ISA) плат расширения с интегрированными контроллерами: последовательного порта (2x16550), параллельного порта, игрового порта, НГМД и НЖМД. Причем все эти функции сочетались в одной БИС с типичным названием Multi I/O.

Основными производителями чипов Multi I/O были компании Winbond, UMC, GoldStar и другие.



[Gigagon]

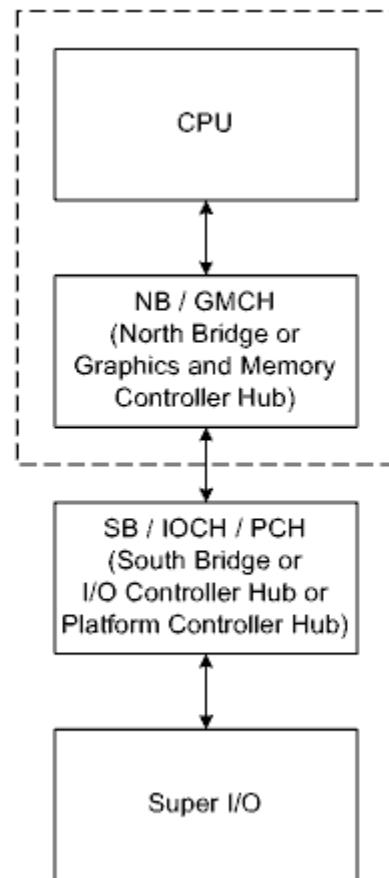
2.0.1.5

Для ПК на базе поздних Intel486 уже была характерна интеграция чипа Multi I/O на материнскую плату.

Одними из ведущих производителей таковой элементной базы стали компании SiS и OPTi. Компания Intel постепенно отказалась от производства микросхем интегрированной периферии и сосредоточилась на разработке наборов микросхем («чипсетах») системной логики (последней БИС Multi I/O, которая была выпущена самой компанией Intel, стала 82091).

2.0.1.6а

5. Во времена процессоров Pentium сформировалась действительная до сих пор базовая крупноблочная структура материнской платы ПК, состоящая из четырех основных БИС.



2.0.1.6b

Контроллеры последовательного порта (по той же схеме 2x16550) в составе интегрированной периферии были перенесены и в эту структуру.

Однако, в связи с некоторой заменой функционала интегрированной периферии (например, удаление контроллера НЖМД и добавление контроллера клавиатуры), вместо названия Multi I/O стало больше использоваться название Super I/O. С этого момента реализации последовательных портов не претерпели никаких изменений.

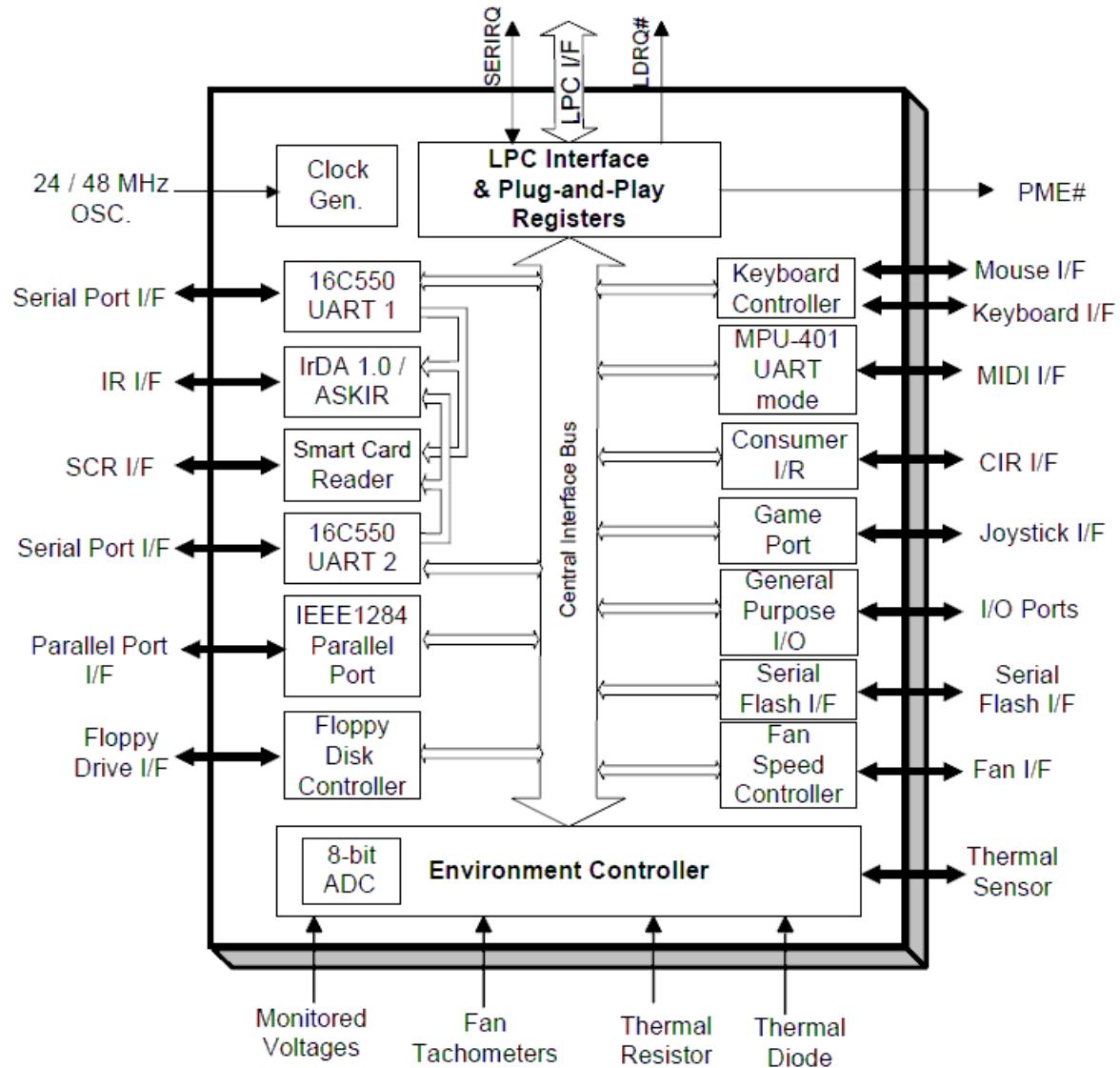
Основными производителями чипов Super I/O являются компании Winbond, ITE и SMSC.



2.0.1.6с

После перехода от мостовой (bridges) организации ПК к хабовой (hubs) в рамках данной структуры (начиная с восьмисотой серии чипсетов Intel в эпоху Pentium III) для внутрисхемного подключения Super I/O вместо шины X-Bus стала использоваться шина LPC (Low Pin Count) -- специализированная разновидность шины PCI с небольшим числом разрядов.

2.0.1.7



Структурная схема Super I/O на примере ITE IT8712F [ITE]

2.0.1.8

6. В настоящее время (приблизительно с 2005 года) традиционный последовательный интерфейс ПК считают устаревшим (legacy), часто исключают из состава интегрированной периферии -- на материнских платах можно увидеть все реже.

Однако возобновлено производство мультикарт -- новые версии представляют собой платы расширения с интерфейсом PCI.

2.0.1.9

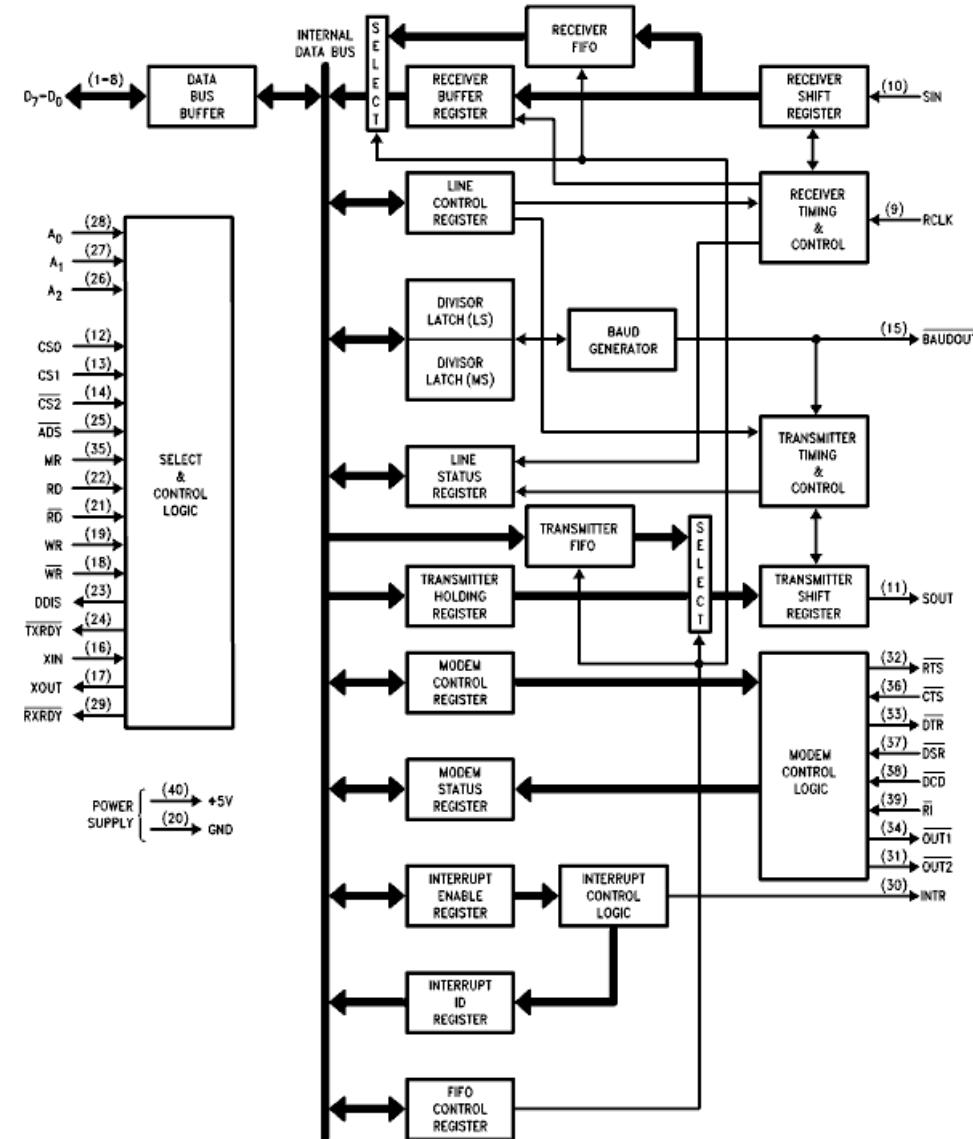
Сейчас в качестве основного последовательного интерфейса ПК рассматривают шину USB (Universal Serial Bus), впервые введенную в состав ПК еще в эпоху процессоров Pentium.

2.0.2.1

Сам факт передачи информации подразумевает наличие передатчика, приемника и канала, по которому они связаны. Как и следует из названия, UART 16550 сочетает в себе функции как приемника, так и передатчика. Предоставлена возможность подключения к двунаправленному **физическому** каналу связи (**или, по-другому, линии**) в соответствии со стандартом RS-232.

На аппаратном уровне приемник и передатчик работают параллельно, то есть по отдельным физическим цепям полностью независимо друг от друга.

2.0.2.2



Структурная схема UART 16550 [National Semiconductor]

2.0.2.3

Интерфейс RS-232 (традиционное название, последнюю редакцию 1997 года правильно называть TIA-232-F, существуют и другие названия) предназначен для подключения АПД (например, модема) к ООД (например, UART).

Для физического подключения по стандарту RS-232 используют девятиконтактные разъемы D Subminiature (D-sub) DE-9. В старых ПК класса IBM PC использовали и аналогичные двадцатипятиконтактные разъемы DB-25.

Принято, что штыревую часть разъема устанавливают со стороны ООД, а гнездовую часть -- со стороны АПД.



Логотип последовательного порта: «|○○|».

Согласно PC System Design Guide, с 1999 года разъемы последовательных портов окрашивают в **бирюзовый цвет**.

2.0.2.4

Традиционное назначение цифровых цепей RS-232:

- SOUT (Serial Output) -- выход передатчика;
- SIN (Serial Input) -- вход приемника;
- RTS (Request to Send) -- сигнал-запрос от UART к модему о передаче байта;
- CTS (Clear to Send) -- сигнал-подтверждение от модема к UART о готовности принять байт для передачи;
- DSR (Data Set Ready) -- сигнал от модема к UART о готовности к взаимодействию;
- DTR (Data Terminal Ready) -- сигнал от UART к модему о готовности к взаимодействию;
- DCD (Data Carrier Detect) -- сигнал от модема к UART об обнаружении данных;
- RI (Ring Indicator) -- сигнал от модема к UART об обнаружении входящего телефонного звонка.

2.0.2.5а

Служебные цепи RS-232 позволяют организовать контроль информационного потока (flow control). Например, это позволяет избегать переполнения приемника, приостанавливая «быстрый» передатчик.

Следует отметить, что практически все служебные цепи напрямую связаны с соответствующими регистрами управления и состояния UART 16550, то есть «открыты» для программирования. Следовательно, алгоритмы контроля реализуют программно и закладывают, например, в драйверы операционных систем.

Контроль может быть как полуаппаратным (с задействованием сигналов RS-232), так и сугубо программным.

Очевидно, что традиционное использование пары RTS/CTS позволяет контролировать передачу только в одном направлении -- от UART к модему. Для контроля передачи в обратном направлении использовалась пара DSR/DTR.

2.0.2.5b

В большинстве современных реализаций контроль по прежнему предполагает наличие обратной связи, но осуществляется только приемником. Два основных метода:

1. RTS/CTS -- полуаппаратный.
2. XON/XOFF -- программный.

UART контролирует передачу данных «к себе» управляя активностью цепи RTS, modem -- CTS.

Значительно реже применяют метод DTR/DSR -- полностью аналогичен методу RTS/CTS, но значения сигналов сохраняются на протяжении всего информационного обмена, а не каждой посылки.

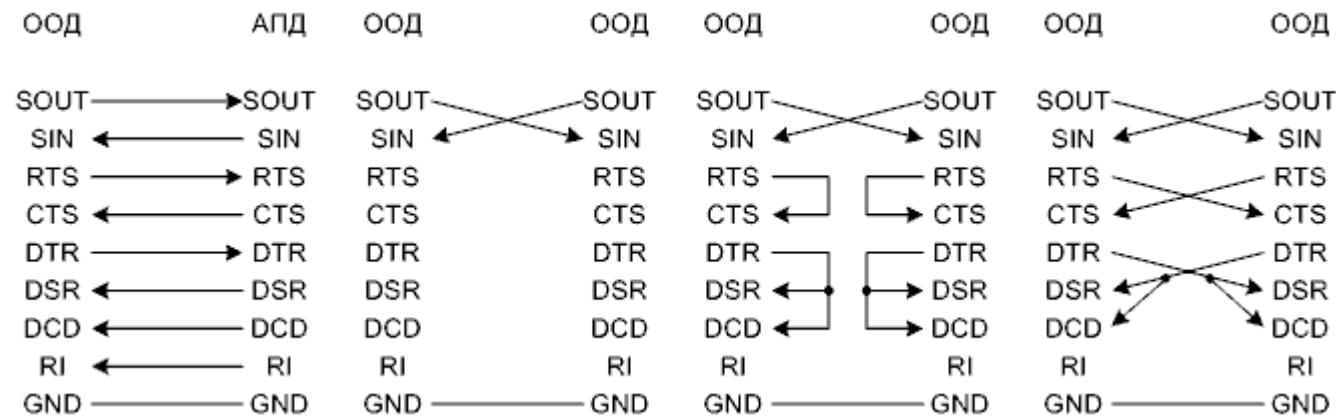
При полностью программном контроле, приемник передает в обратном направлении специальный байт XON (стандартное значение 11h) для инициирования передачи и специальный байт XOFF (стандартное значение 13h) для остановки передачи.

2.0.2.6а

В стандартной ситуации, ООД взаимодействуют между собой посредством АПД, причем с помощью так называемых «рукопожатий» (handshaking) с АПД. При этом подключение АПД к ООД осуществляют посредством «прямого» кабеля (straight-through cable).

Для подключения двух ООД друг к другу непосредственно необходим один из вариантов нуль-модемного (null-modem, поскольку предполагают отсутствие модема) кросс-кабеля (crossover cable, поскольку цепи SIN и SOUT скрещивают).

2.0.2.6b



Для изготовления кросс-кабеля нужны минимум три провода.

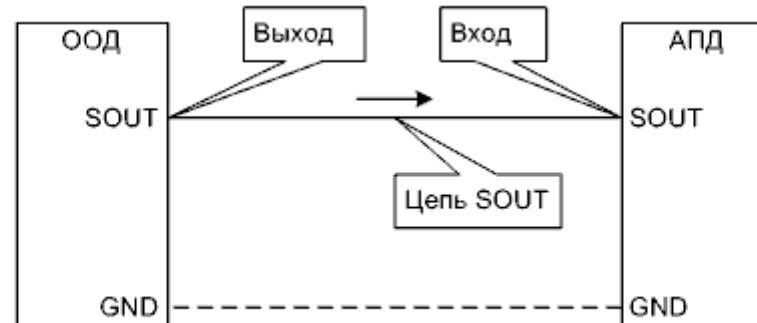
Иногда программное обеспечение рассчитано только на использование модемов. В подобных ситуациях, для непосредственного подключения двух ООД необходимо «закоротить» соответствующие пары сигналов.

Часто в литературе приводят еще одну схему нуль-модемного кабеля, в соответствии с которой роль модема играет ООД-абонент.

2.0.2.7

Скрещивание, как таковое, обусловлено необходимостью правильного согласования электрических цепей.

При этом нужно иметь в виду, что в информационных системах принято все «мерить» относительно человека, в том числе и ввод-вывод. Таким образом, в ПК направление определяется «с точки зрения» центрального процессора, а в данном случае -- «с точки зрения» ООД.



2.0.2.8

Как известно, без синхронизации в том либо ином виде передать ничего не возможно. Передатчик и приемник, по крайней мере, должны «встретиться во времени». Поскольку передатчик и приемник не имеют общего источника времени, в **линию** вводятся специальные синхросигналы.

С точки зрения синхронизации, применительно к последовательным каналам связи выделяют два режима обмена:

1. Асинхронный (asynchronous) -- синхронизируется посылка каждого информационного байта.
2. Синхронный (synchronous) -- синхронизируется весь информационный обмен.

2.0.2.9a



Формат посылки в асинхронном режиме

2.0.2.9b

Атомарной, то есть минимальной неделимой единицей, с которой работает как UART, так и USART, является байт, причем один байт не обязательно равен восьми битам и может содержать от 5 до 8 битов.

По умолчанию линия находится в состоянии логической единицы.

При наличии байта для передачи передатчик переводит линию в состояние логического нуля, то есть передает старт-бит, что говорит приемнику о том, что на следующем такте нужно «ловить» первый информационный бит.

Стоп-бит необходим для того, чтобы после передачи информационной последовательности гарантированно вернуть линию в исходное, то есть единичное состояние.

Старт-бит всегда один, а стоп-битов может быть один, полтора либо два.

2.0.2.9с

Для проверки целостности информационной части, если эта проверка включена, за информационной частью вставляется бит паритета.

При этом действует правило дополнения. Например, если включена проверка единиц на четность (even), то бит паритета формируется таким образом, чтобы общее число единиц (в информационной части плюс бит паритета) было четным. Либо, если включена проверка нулей на нечетность (odd), то общее количество нулей должно быть нечетным.

Ошибки отслеживаются приемником.

2.0.2.10a



Формат посылки в синхронном режиме

2.0.2.10b

При «простое» передатчик **«заполняет»** линию специальными байтами синхронизации, тем самым настраивая приемник.

Все поступающие байты передаются без «обрамления».

Как и в асинхронном режиме, ошибки отслеживаются приемником. При обнаружении ошибок, а при длительной непрерывной передаче из-за накапливающихся фазовых сдвигов они неизбежно возникают, приемник должен каким-либо дополнительным способом (так как текущую **цепь** задействовать невозможно) приостановить передатчик, чтобы **ТОТ ВНОВЬ «заполнил» линию** байтами синхронизации.

2.0.2.11

По своей сути, передатчик и приемник СОМ-порта представляют собой программируемые сдвиговые регистры.

Данные, предварительно записанные в регистр передатчика параллельно, затем последовательно сдвигаются в **линию** под воздействием тактовых импульсов. В процессе работы UART 16550 тактирование сдвиговых регистров осуществляется непрерывно. Следовательно, данные начинают поступать в **линию** сразу после их записи в регистр передатчика.

Заполнение регистра приемника так же происходит «автоматически». Если передаваемые байты записываются слишком быстро, то возникает переполнение очереди FIFO передатчика. Если принимаемые байтычитываются слишком медленно, то после переполнения очереди FIFO приемника происходит их потеря.

2.0.2.12

Тактирование сдвиговых регистров UART 16550 осуществляется с помощью встроенного программируемого бод-генератора (baud generator) (тактирование некоторых первых реализаций UART осуществлялось таймером).

Бод-генератор представляет собой программируемый делитель частоты.

Выходная частота бод-генератора F_{out} определяется по формуле:

$$F_{out} = F_{in} / (16 \cdot DL),$$

где:

F_{in} -- входная частота,

DL -- шестнадцатибитная константа, старшая и младшая части которой хранятся в двух регистрах UART (DLL и DLM).

2.0.2.13

На вход бод-генератора поступает меандр, получаемый от внешнего кварцевого резонатора, который тактирует и сам автомат UART. Частота тактирования автомата UART по крайней мере в 16 раз больше F_{out} .

Следует учитывать, что, для того чтобы правильно рассчитать DL , необходимо точно знать F_{in} .

Вполне естественно, что на разных материнских платах используют разные микросхемы и разные кварцевые резонаторы. Применительно к современным Super I/O, эта частота может достигать 48 MHz, то есть совпадать с частотой синхронизации Super I/O.

Но, за счет еще одного деления частоты (при загрузке ПК BIOS конфигурирует UART инициализируя соответствующие регистры конфигурационного пространства Super I/O), как правило, F_{in} приводится к классическому значению 1,843 MHz.

При этом, если $DL = 1$ (нулевое значение DL использовать крайне не рекомендуется), то $F_{out} = 115200$ Hz.

2.0.2.14

Пропускную способность последовательных каналов связи принято оценивать в бодах.

Один бод (baud) равен одному сигналу в секунду.

В случае с UART 16550 производительность, измеренная в бодах, совпадает с производительностью, измеренной в битах в секунду (bit/s равно bps).

2.0.2.15

UART 16550, как и вся БИС Super I/O, как и любая БИС на материнской плате, является низковольтной.

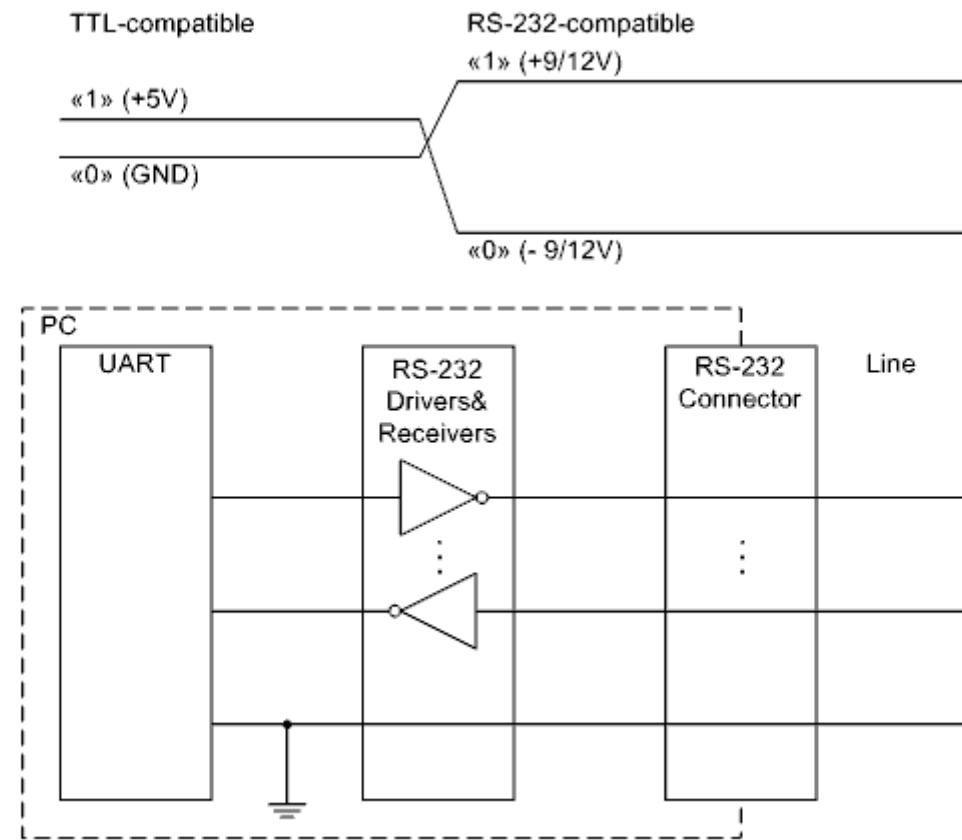
Но в интерфейсе RS-232 значения логических уровней совершенно другие, значительно более «разнесенные», что позволяет передавать данные на расстояние до нескольких десятков метров.

Для получения необходимых значений используют специализированные преобразователи уровней 75232 (аналоги: 75185, 6571 и другие).



2.0.2.16

Преобразователь уровней 75232 фактически играет роль *трансивера* (transceiver, transmitter плюс receiver), сочетаая функции приемника и передатчика в интерфейсе с определенной физической средой, которой в данном случае является RS-232).



2.0.2.17

В больших ЭВМ производства второй половины прошлого века для подключения терминалов применялись другие трансиверы -- трансиверы токовой петли (current loop), которые не «разносят» уровни напряжений, а моделируют токовые посылки, что позволяет увеличить расстояние передачи.

Возможность использования токовой петли в ПК была отвергнута изначально.

2.0.3.1a

Как и любое устройство ввода-вывода, UART 16550 содержит регистры управления, регистры состояния, плюс информационные регистры.

В стандартной архитектуре ПК для COM1 и COM2 зарезервированы следующие диапазоны программных портов в адресном пространстве ввода-вывода процессора: 3F8h -- 3FFh и 2F8h -- 2FFh соответственно (но возможности Super I/O позволяют сконфигурировать UART 16550 нестандартно).

2.0.3.1b

Регистры UART 16550 отображаются в соответствующий диапазон следующим образом.

Register Address Access (AEN = 0)		Abbreviation	Register Name	Access
Base +	DLAB			
0h	0	THR	Transmit Holding Register	WO
0h	0	RBR	Receiver Buffer Register	RO
0h	1	DLL	Divisor Latch LSB	R/W
1h	1	DLM	Divisor Latch MSB	R/W
1h	0	IER	Interrupt Enable Register	R/W
2h	—	IIR	Interrupt Identification Register	RO
2h	—	FCR	FIFO Control Register	WO
3h	—	LCR	Line Control Register	R/W
4h	—	MCR	Modem Control Register	R/W
5h	—	LSR	Line Status Register	R/W
6h	—	MSR	Modem Status Register	R/W
7h		SCR	Scratch Pad Register	R/W

(Регистры данных, как и цепи для передачи и приема данных, в разных документах называют по-разному.)

Отображение частично зависит от значения Divisor Latch Access Bit (DLAB) -- самого старшего (седьмого) бита регистра LCR.

2.0.3.2

Прикладная программа должна в первую очередь корректно инициализировать соответствующие регистры UART.

При этом представлена возможность работы по прерываниям.

Стандартными аппаратными прерываниями COM1 и COM2 являются IRQ4 и IRQ3 соответственно (также можно изменить).

2.0.3.3

Назначение регистров:

1. THR (Transmit Holding Register) -- регистр данных передатчика (точнее буферный регистр сдвигового регистра передатчика).
2. RBR (Receiver Buffer Register) -- регистр данных приемника (точнее буферный регистр сдвигового регистра приемника).
3. DLL (Divisor Latch Least significant byte) -- младшая часть константы деления бод-генератора.
4. DLM (Divisor Latch Most significant byte) -- старшая часть константы деления бод-генератора.

2.0.3.4

5. IER (Interrupt Enable Register) -- регистр разрешения прерываний.



2.0.3.5

6. IIR (Interrupt Identification Register) -- регистр идентификации прерываний.



2.0.3.6

Бит 3 IIR	Бит 2 IIR	Бит 1 IIR	Бит 0 IIR	Приоритет прерывания	Тип прерывания	Причины прерывания	Условие сброса прерывания
0	0	0	1	Нет	Нет	Нет	Нет
0	1	1	0	Первый (наивысший)	Статус линии изменился	Переполнение, ошибка паритета, ошибка в формате посылки	Чтение LSR
0	1	0	0	Второй	Приемник заполнен	Данные успешно приняты	Чтение RBR
1	1	0	0	Второй	Тайм-аут	Не было чтения и записи в очередь FIFO приемника в течение времени, соответствующего приему четырех байтов, и в очереди находится по крайней мере один байт	Чтение RBR
0	0	1	0	Третий	Передатчик пуст	Данные успешно переданы	Чтение IIR или запись в THR
0	0	0	0	Четвертый	Статус модема изменился	CTS, DSR, RI, DCD	Чтение MSR

Прерывания UART 16550

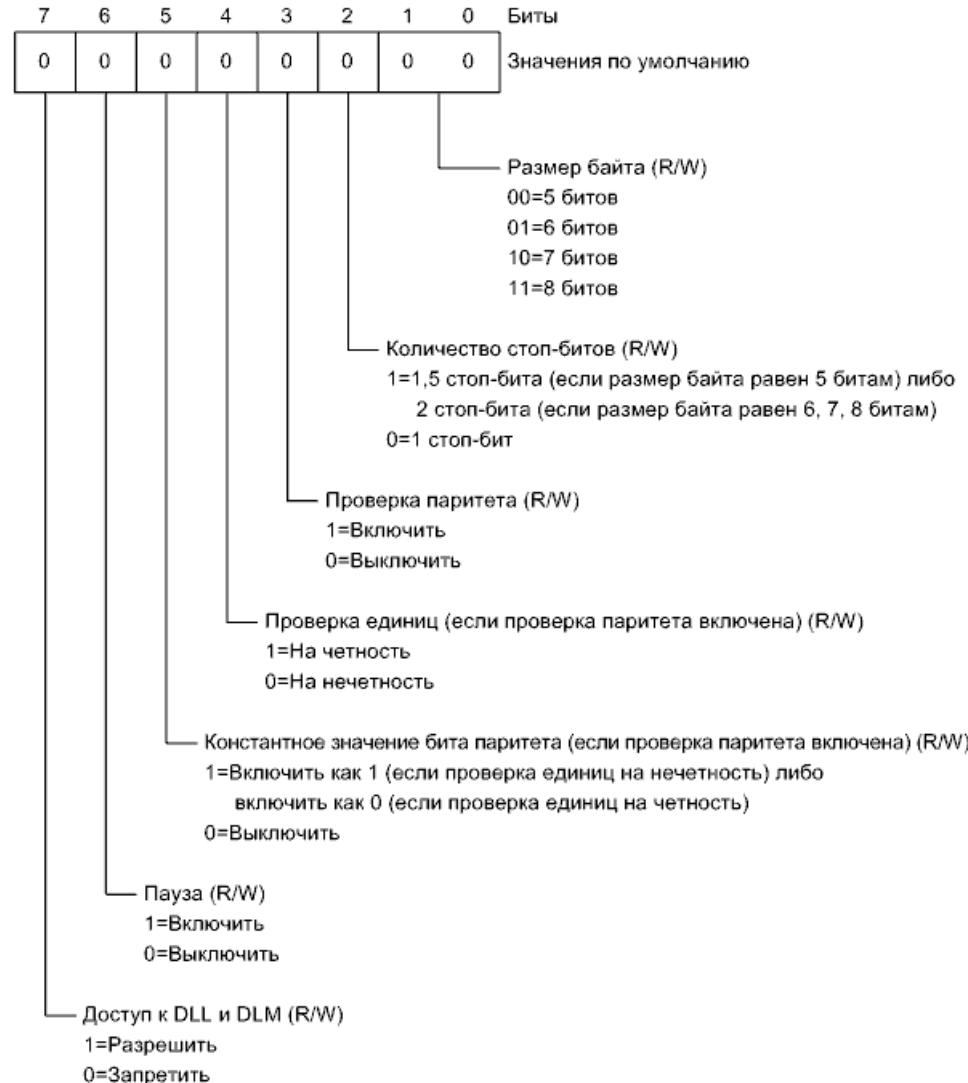
2.0.3.7

7. FCR (FIFO Control Register) -- регистр управления очередями FIFO передатчика и приемника.



2.0.3.8a

8. LCR (Line Control Register) -- регистр управления линией.



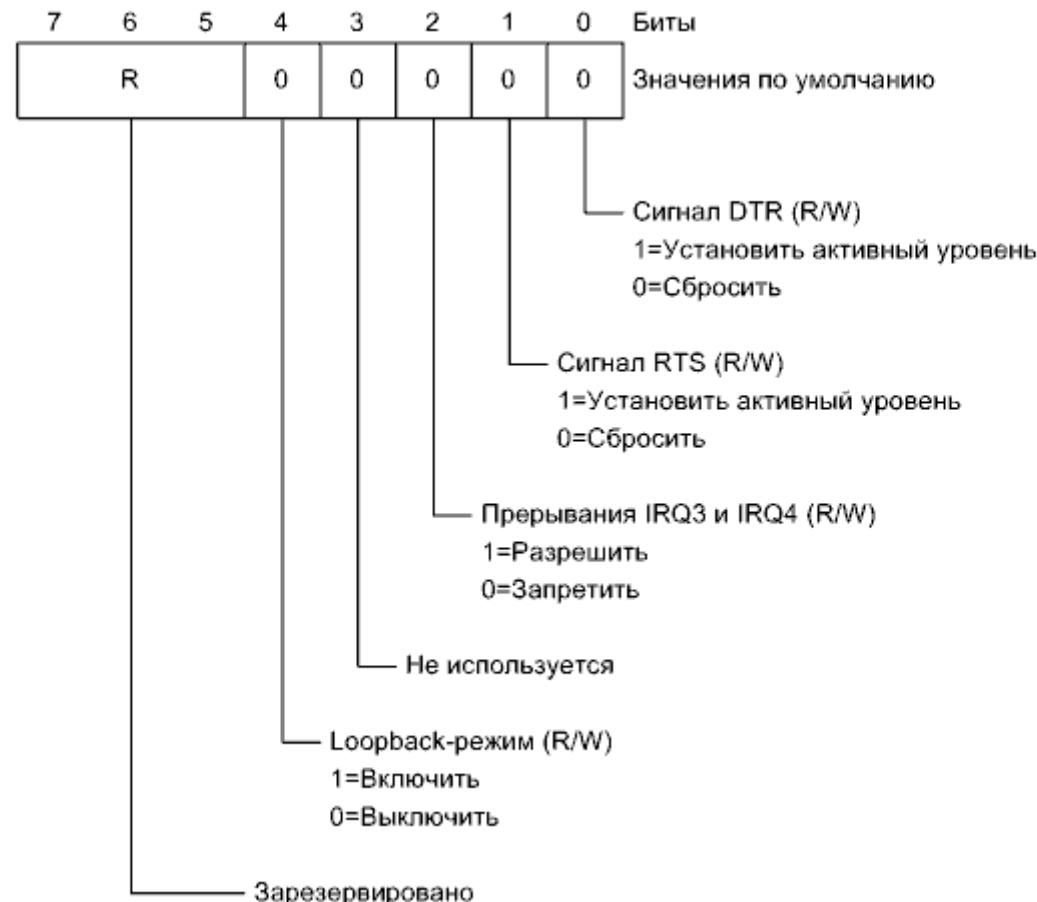
2.0.3.8b

Включение «залипания» бита паритета (sticky parity) приводит к передаче соответствующего константного значения.

Включение паузы приводит к приостановке передатчика. При этом передатчик удерживает линию в состоянии логического нуля длительное время, что автоматически переводит в режим паузы и приемник (без уведомления об ошибках).

2.0.3.9a

9. MCR (Modem Control Register) -- регистр управления модемом.



2.0.3.9b

Включение loopback-режима приводит к «закорачиванию» выхода передатчика и входа приемника, что может применяться с целью тестирования UART.

2.0.3.10a

10. LSR (Line Status Register) -- регистр состояния линии.



2.0.3.10b

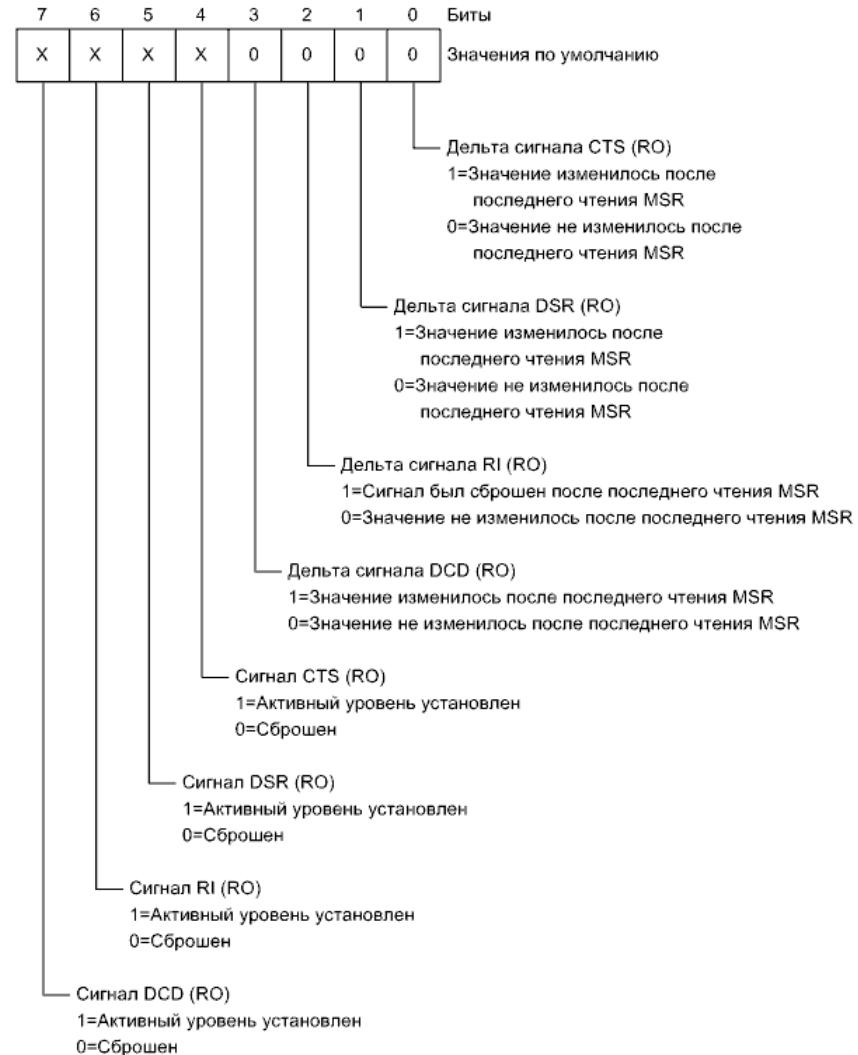
После считывания очередных данных из приемника нулевой бит LSR обнуляется.

После записи очередных данных в передатчик обнуляются пятый и шестой биты LSR.

Остальные биты обнуляются после чтения LSR.

2.0.3.11

11. MSR (Modem Status Register) -- регистр состояния модема.



2.0.3.12

12. SCR (Scratch Pad Register) -- дополнительный регистр для временного хранения данных, не связанный с функционированием UART.

2.0.4.1

С точки зрения топологии, интерфейс RS-232 обладает одним существенным ограничением, которое закономерно вытекает из его природы. Он изначально задумывался как интерфейс между разноранговыми устройствами, то есть, по сути дела, как интерфейс для подключения периферийных устройств к компьютеру. Более двух устройств с помощью RS-232 объединить невозможно.

В результате, закономерным продолжением стандарта RS-232 стали два стандарта: RS-422 (EIA-422-B) и RS-485 (EIA-485). При этом RS-422 можно рассматривать как промежуточный на пути к RS-485 стандарт.

2.0.4.2

Характеристика	RS-232	RS-422	RS-485
Способ передачи сигнала	Изменение потенциала относительно земли	Дифференциальная пара	Дифференциальная пара
Направление передачи	Одностороннее, двустороннее	Одностороннее, двустороннее	Одностороннее, двустороннее
Максимальное количество передатчиков	1	1	32
Максимальное количество приемников	1	10	32
Ориентировочная максимальная пропускная способность	1 Mbit/s	10 Mbit/s	10 Mbit/s
Ориентировочное максимальное расстояние	15 м	1200 м	1200 м

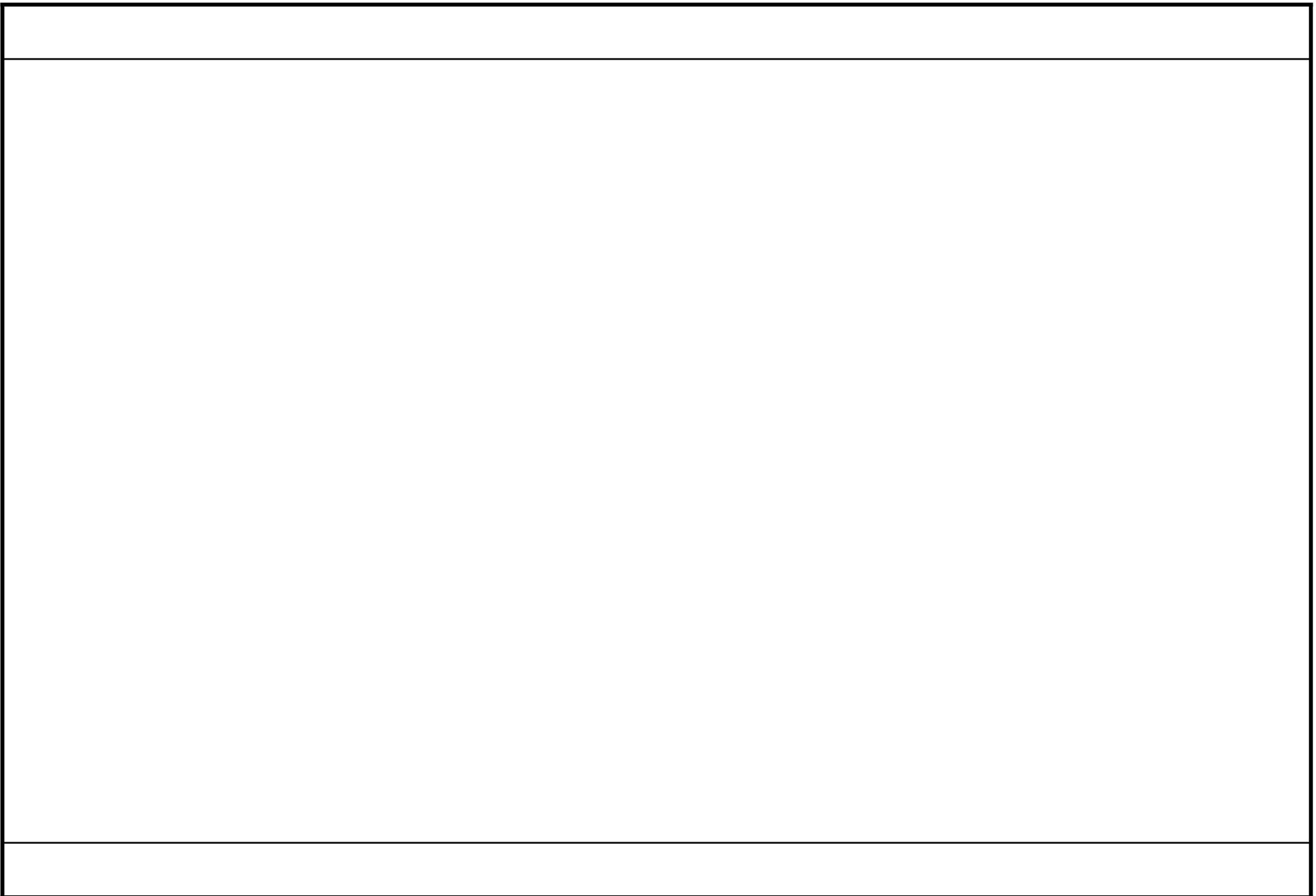
Основные сравнительные характеристики RS-232, RS-422 и RS-485

2.0.4.3

Для передачи данных посредством интерфейса RS-485 требуются специальные трансиверы с гальванической развязкой, позволяющие реализовать дифференциальный способ передачи сигнала.

Гальваническая развязка может быть либо трансформаторной, либо оптронной.

О СрПД в стандарте не сказано, но, как правило, используют витую пару (twisted pair) и разъемы типа RJ.



ПАКЕТНАЯ ПЕРЕДАЧА ДАННЫХ

3.0.1.1

Исторически так сложилось, что компьютерные сети имеют последовательную природу. Объяснить это можно тем, что реализовать передачу данных на сравнительно больши'e расстояния в параллельном виде значительно сложнее, чем в последовательном. Между станциями данные передаются по последовательным каналам, а внутри станций обрабатываются параллельно.

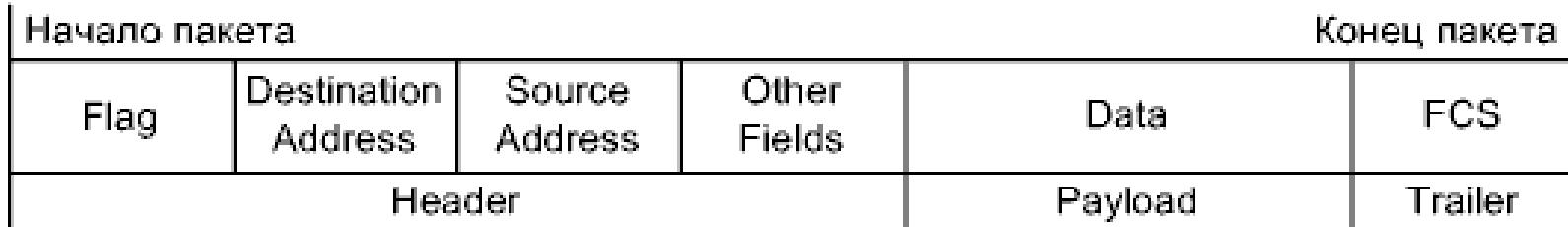
3.0.1.2

Для именования порции информации, передаваемой по каналам компьютерных (и не только компьютерных) сетей, используют обобщенный термин *пакет* (packet).

Пакет содержит последовательно сформированные станцией-передатчиком поля (fields), предназначенные для их интерпретации в станции-приемнике.

В общем случае, пакеты могут быть самыми разнообразными (как по структуре, так и по длине), но подавляющее большинство пакетов подпадают под типовую структуру.

3.0.1.3



Назначение полей:

Flag -- флаг, точнее, флаг начала пакета -- позволяет определить начало пакета.

Destination Address -- адрес назначения -- позволяет указать станцию, для которой предназначен пакет.

Source Address -- адрес источника -- позволяет указать станцию, сгенерировавшую пакет.

Other Fields -- прочие поля -- специфические поля (в том числе и специфические флаги) определенной реализации.

Data -- данные -- «полезное» наполнение пакета.

FCS (Frame Check Sequence) -- контрольная сумма -- позволяет проверить целостность пакета.

Структура типового пакета КС

3.0.1.4

Часть пакета, включающую поля, расположенные до начала данных, принято называть *заголовком* (header) пакета, после данных -- *хвостовиком* (trailer).

3.0.1.5

Обычно в байт-ориентированных реализациях длина пакета кратна восьми битам, то есть пакет состоит из так называемых октетов (octets).

При изображении структуры пакетов старшие разряды принято располагать слева или сверху (most significant bit first, big endian).

В процессе передачи, поля сдвигаются в канал по очереди, то есть начиная с левого поля.

Если поле состоит из нескольких октетов, то октеты как правило так же сдвигаются начиная с левого октета.

А вот биты октетов в реализациях сдвигаются по-разному -- как начиная с левого бита (основной вариант в семействе протоколов TCP/IP), так и начиная с правого бита (основной вариант в Ethernet); даже может быть, что биты октетов разных полей сдвигаются по-разному.

3.0.1.6

Все поля в составе любого пакета можно условно разделить на полезные и служебные.

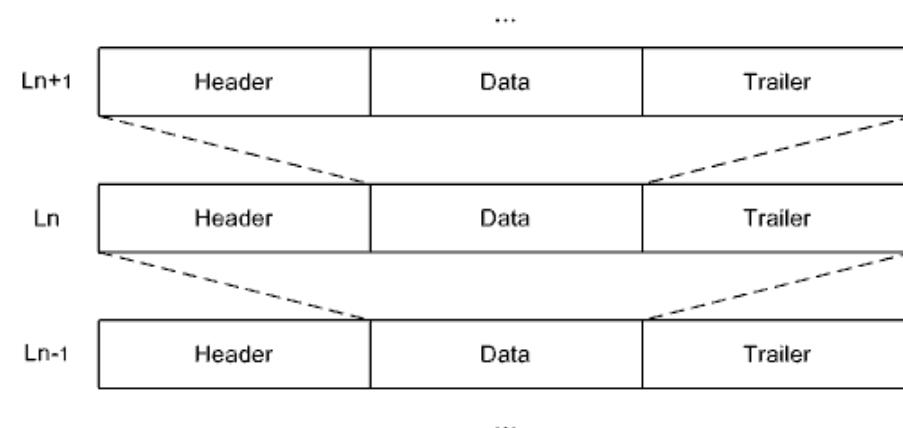
Полезная нагрузка (payload) заключается в собственно данных. Но следует понимать, что вкладываемая в качестве данных информация может носить служебный характер. В некоторых пакетах поле данных не предусмотрено вообще.

Сколько дополнительного трафика порождается в связи с наличием служебных полей оценивают как overhead.

3.0.2.1

В соответствии с концепцией модели OSI, соседние уровни абстрагированы друг от друга. Поэтому вполне закономерно, что на каждом уровне работают со своими структурами данных. При продвижении информации между уровнями возникает необходимость в преобразованиях структур данных. Преобразования выражаются в инкапсуляции и декапсуляции.

Под *инкапсуляцией* (encapsulation) в КС понимают вкладывание пакета определенного вышестоящего уровня в поле данных пакета смежного нижестоящего уровня в процессе подготовки к передаче, то есть при продвижении сверху вниз. Под *декапсуляцией* (decapsulation) понимают обратное действие после приема, то есть при продвижении снизу вверх.



3.0.2.2

Приведите пример инкапсуляции.

3.0.2.3

Как происходит декапсуляция?

3.0.2.4

Функционал любого из вышестоящих уровней «знает», какие нижестоящие ресурсы ему необходимы и чем он «располагает». Поэтому процесс инкапсуляции не доставляет трудностей.

А вот функционал нижестоящего уровня при разборе полученных пакетов заранее не знает, какой из вышестоящих подсистем передавать эти пакеты.

Проблему решают введением в структуру пакета служебного поля, в котором записывается код протокола вышестоящего уровня.

3.0.2.5

Ethertype	Ethertype (hex)	Description
(decimal)		
0000	0000-05DC	IEEE802.3 Length Field
0257	0101-01FF	Experimental
0512	0200	XEROX PUP (see 0A00)
0513	0201	PUP Addr Trans (see 0A01)
	0400	Nixdorf
1536	0600	XEROX NS IDP
	0660	DLOG
	0661	DLOG
2048	0800	Internet Protocol version 4 (IPv4)
2049	0801	X.75 Internet
2050	0802	NBS Internet]
2051	0803	ECMA Internet
2052	0804	Chaosnet
2053	0805	X.25 Level 3
2054	0806	Address Resolution Protocol (ARP)
2055	0807	XNS Compatability
2056	0808	Frame Relay ARP
2076	081C	Symbolics Private
2184	0888-088A	Xplex
2304	0900	Ungermann-Bass net debugr
2560	0A00	Xerox IEEE802.3 PUP
2561	0A01	PUP Addr Trans
2989	0BAD	Banyan VINES
2990	0BAE	VINES Loopback
2991	0BAF	VINES Echo
4096	1000	Berkeley Trailer nego
4097	1001-100F	Berkeley Trailer encap/IP
5632	1600	Valid Systems
	22F3	TRILL
	22F4	L2-IS-IS
...		
	86DD	Internet Protocol version 6
...		
65535	FFFF	Reserved

IEEE 802 Numbers [IANA]

3.0.2.6

Как вы думаете, что такое туннелирование (tunneling)?

3.0.2.7

Важной особенностью инкапсуляции является то, что в большинство реализаций заложена возможность передавать пакеты, относящиеся к некоторому протоколу некоторого уровня (например, сетевого), вкладывая их в пакеты другого протокола того же уровня, то есть организовывать *туннелирование* (tunneling).

3.0.2.8

Для чего может применяться туннелирование?

3.0.2.9

Инкапсуляция имеет еще ряд проявлений.

Если при выполнении инкапсуляции данные некоторого уровня не помещаются в поле отведенной длины, то можно прибегнуть к *фрагментации* (fragmentation) -- разбить данные на фрагменты и передать цепочку пакетов. Принимающая сторона будет вынуждена выполнить *дефрагментацию* (defragmentation).

Поле, отвечающее за длину поля данных, может быть не предусмотрено. Если длина поля данных фиксирована, а данных не хватает, то возникает необходимость в автодополнении (например, нулями).

3.0.2.10

Перемежение (interleaving) позволяет «распараллелить» пересылку пакетов или их фрагментов и заключается в одновременном задействовании нескольких каналов.

Особенно это применимо в низкоскоростных СрПД.

3.0.2.11

Фрагментация (при наличии альтернативных путей в СПД) и перемежение могут привести к «перемешиванию» пакетов и, как следствие, разрушению сообщения.

Контроль за порядком фрагментов может быть возложен как на протокол подверженного фрагментации уровня, так и на протокол вышестоящего уровня.

3.0.3.1

Несмотря на целостность уровней, вышестоящие уровни зависят от нижестоящих. Но степень зависимости различается.

Иногда требуется просто наличие поддержки одного из нижестоящих протоколов, иногда требуется поддержка конкретного нижестоящего протокола.

3.0.3.2

Названия структурных единиц передаваемой информации в привязке к уровням модели OSI:

L1 -- сигналы (signals).

L2 -- кадры (frames).

L3 -- собственно пакеты (packets).

L4 + L5 -- сегменты (segments).

L6 + L7 -- сообщения (messages).

Фундаментальная задача СПД заключается в том, чтобы правильно передать сообщение.

Пакеты «возникают» начиная со второго уровня, хотя собственно пакетами традиционно называют пакеты, относящиеся к третьему уровню.

Первый и второй уровни часто совмещают в рамках аппаратных технологий.

Четвертый и пятый уровни, равно как шестой и седьмой уровни, обычно реализуют «неразрывно» в рамках программных технологий.

Для обобщенной ссылки на порцию данных, над которой оперируют на некотором уровне, Cisco использует термин PDU (Protocol Data Unit).

3.0.4.1

Понятно, что для правильной интерпретации пакета нужно его считать из канала полностью, причем с соблюдением последовательности. Если бы взаимодействующие станции работали бесконечно и находились в соответствующей степени готовности, то это не составляло бы особого труда. Но, поскольку станция-приемник может подключиться к каналу (да и вообще начать работать) в произвольный момент времени, возникает проблема, связанная с распознаванием флага начала пакета.

Флаг начала пакета представляет собой зарезервированную цифровую последовательность, которая собственно позволяет станции-приемнику определить начало пакета.

Проблема заключается в том, что такая же последовательность вполне может встретиться в пакете и после флага начала. Следовательно, возникает задача обеспечения уникальности флага начала пакета, то есть исключения этой последовательности из оставшейся части пакета.

Это достигается за счет действия, заключающегося в модификации следующей за флагом цифровой последовательности, которое в бит-ориентированных системах называют **бит-стаффингом** (bit stuffing), а в байт-ориентированных -- **байт-стаффингом** (byte stuffing).

3.0.4.2

При бит-стаффинге совпадающая с флагом последовательность разбивается с помощью вставки дополнительного бита с соответствующим значением.

Применение бит-стаффинга приводит к увеличению длины пакета. Теоретически, с целью уменьшения связанных с бит-стаффингом «издержек», следует стремиться к минимизации количества вставок: разбивающий бит нужно вставлять после наиболее длинной уникальной подпоследовательности в флаговой последовательности.

3.0.4.3

Классическим флагом начала пакета является байт со значением 01111110_b (7Eh).



На передающей стороне после нуля и шести единиц всегда вставляется седьмая единица, а на принимающей стороне единица после нуля и шести единиц всегда удаляется.

3.0.4.4

Цель байт-стаффинга полностью совпадает с целью бит-стаффинга. В сравнении с алгоритмами бит-стаффинга, алгоритмы байт-стаффинга манипулируют байтами, являются более сложными и более «затратными», но при программировании они позволяют избежать битовых операций (бит-стаффинг, в отличие от байт-стаффинга, обычно реализуют аппаратно).

3.0.4.5



Единственным способом обеспечения уникальности флагового байта является замена совпадающего с ним байта на некий выбранный другой. Но возникает вопрос, как принимающая сторона отличит замененный байт от такого же незамененного. Решением является применение так называемого ESC-символа. Наличие ESC-символа говорит станции-приемнику о факте замены, а следующий за ESC-символом символ -- код замены позволяет определить какая замена была осуществлена. Байт-стаффингу можно подвергать целые группы символов.

3.0.4.6

Бит-стаффинг обычно применяют при задействовании синхронных СрПД, а байт-стаффинг -- асинхронных.

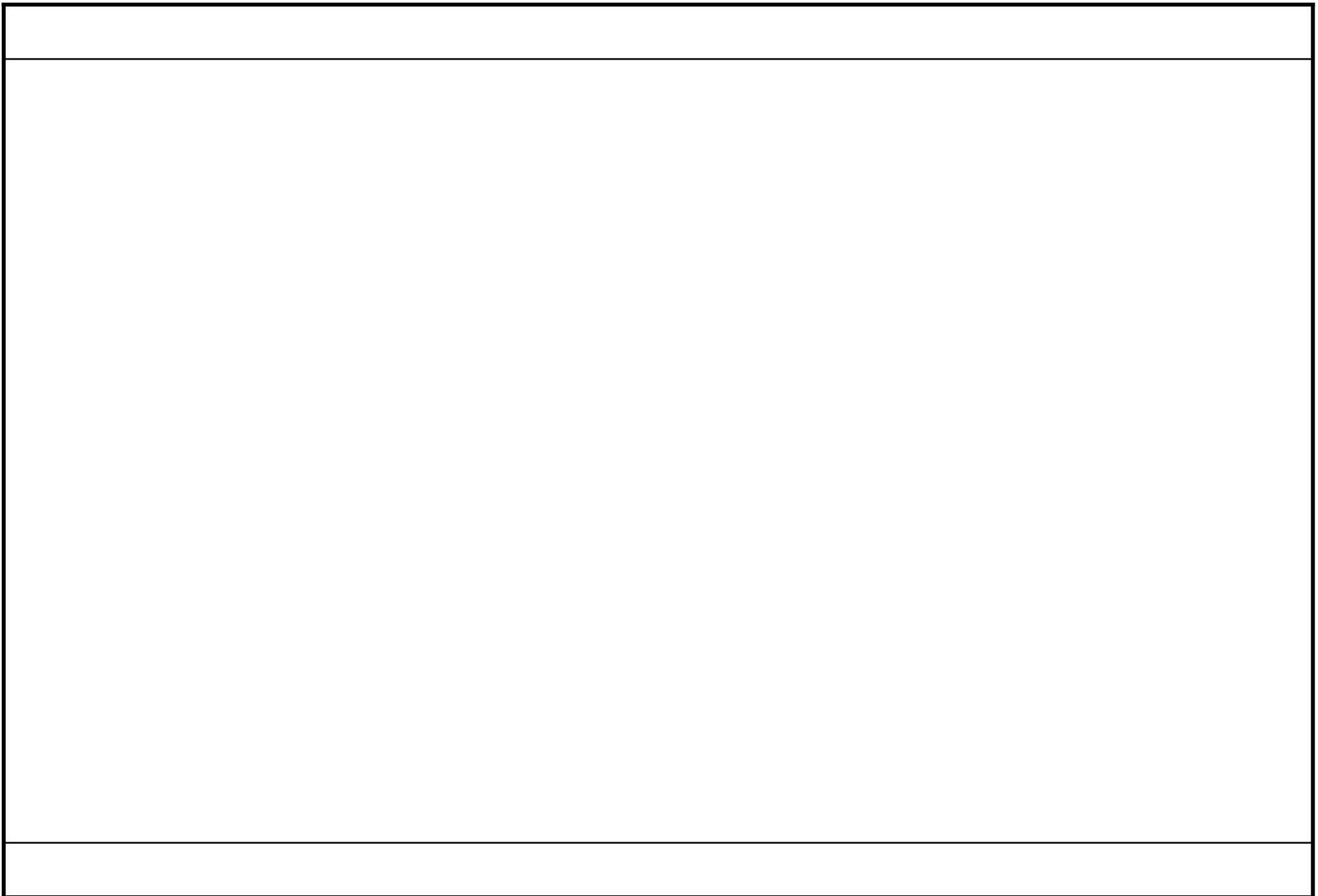
Примерами технологий могут служить SDLC, HDLC, ISDN и другие (многие поддерживают как синхронные так и асинхронные СрПД).

Примером протокола может служить PPP.

Следует отметить, что на практике (например, применительно к HDLC) бит-стаффинг выполняется вставкой нуля после пяти единиц.

3.0.4.7

Нарисуйте принципиальную схему устройства, которое будет выполнять бит-стаффинг и де-бит-стаффинг применительно к стандартному флагу.



КАНАЛЬНОЕ КОДИРОВАНИЕ

4.0.1.1

Кодирование на канальном уровне (канальное кодирование) призвано решать две фундаментальные задачи:

1. Адаптировать битовые последовательности к возможностям физического уровня с целью обеспечения или улучшения требующихся технических характеристик. Лучше всего это назвать *линейным кодированием* (*line encoding*), где слово «линейное» происходит от понятия физической линии.
2. Обеспечить проверку целостности данных и, по возможности, восстановление ошибочных битов. Лучше всего это назвать *помехоустойчивым кодированием* (*antinoise encoding*).

4.0.1.2

Следует отметить, что терминологически линейное кодирование имеет и альтернативный смысл, происходящий от математического понятия линейных функций.

В первом подразделе в этот термин будет вкладываться физический смысл, во втором -- математический.

4.0.1.3

Не следует путать линейное кодирование с модуляцией, выполняемой на физическом уровне. Несмотря на то, что часто эти процессы связаны неразрывно, линейное кодирование все-таки следует рассматривать как надстройку над модуляцией.

4.0.1.4

Фактор помех, если под помехами понимать различные электромагнитные наводки, **в КС безусловно учитывают**. Но борьба с помехами -- это лишь частные случаи обеих задач (даже вторая задача порождена не только воздействием на канал наводок).

4.1

ЛИНЕЙНОЕ КОДИРОВАНИЕ

4.1.1.1

Одной из основных предпосылок для разработки линейных кодов, является проблема, проявляющаяся во многих системах передачи цифровой (не только) информации, известная как *девиацией несущей* (carrier deviation).

Очевидно, передатчик и приемник должны работать на одной частоте. В большинстве случаев, передатчик и приемник имеют разные источники синхронизации. При этом тактовые генераторы далеко не идентичны.

Если состояние линии очень долго не изменяется, что происходит при передаче очень длинных нулевых либо единичных последовательностей с использованием классической амплитудной модуляции цифровых цепей (логический ноль соответствует земле, а логическая единица некоторому положительному потенциалу относительно земли), то приемнику «цепляться не за что». В результате накапливаются фазовые сдвиги, что в конце концов приводит к возникновению ошибок.

Современная схемотехническая база для борьбы с девиацией несущей имеет в распоряжении блок ФАПЧ (фазовой автоподстройки частоты), позволяющий автоматически подстраивать тактовый генератор приемника к тактовому генератору передатчика. Наиболее близкий англоязычный термин -- PLL (Phased-Locked Loop).

4.1.1.2

Все линейные коды, в той или иной степени, направлены на преобразование битовых последовательностей, чтобы в линии постоянно происходили изменения. В том числе, за счет равномерного распределения нулей и единиц.

4.1.2.1

Шесть факторов, влияющих на классификацию линейных кодов:

1. Кодирование уровнями либо переходами.
2. Наличие инвертирования.
3. Однополярность либо многополярность.
4. Наличие так называемого «возврата к нулю».
5. Наличие самосинхронизации.
6. Наличие перестановки или подмены битов.

4.1.2.2

Что такое самосинхронизация?

4.1.2.3

Для изучения в рамках данной дисциплины **выбраны** следующие основные группы кодов:

1. NRZ (Non-Return-to-Zero) codes -- коды без возврата к нулю.
2. RZ (Return-to-Zero) codes -- коды с возвратом к нулю.
3. Manchester codes -- манчестерские коды.
4. MLT (Multi-Level Transmit) codes -- многоуровневые коды.
5. Block codes -- блочные коды.

4.1.3.1

NRZ-коды выражаются в изменении уровней между тактами.

В простейших случаях, логические уровни в исходной последовательности не преобразуются совсем либо инвертируются.

Более сложными случаями являются space и mark. При space-варианте ноль во входной последовательности кодируется сменой текущего уровня в выходной, а единица -- сохранением текущего уровня. При mark-варианте, наоборот, единицы в исходной последовательности приводят к переключению уровней. Начальное состояние значения не имеет.

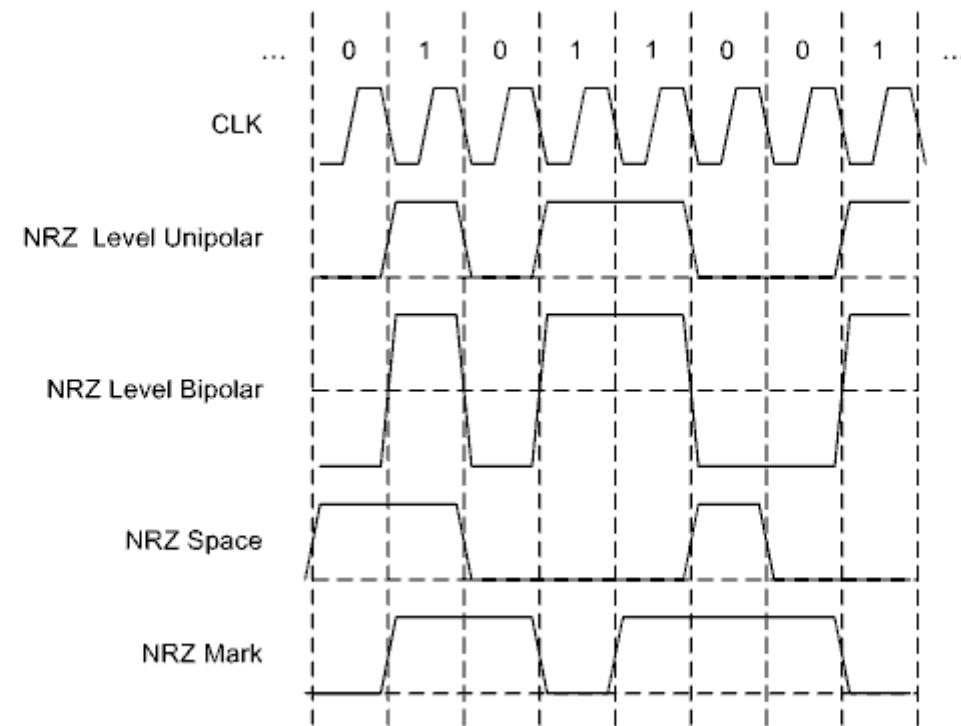
Space и mark инверсны друг относительно друга.

NRZ-коды могут быть однополярными и двухполярными.

Требуется наличие дополнительной цепи для тактирования.

Примеры технологий с применением NRZ-кодов: RS-232, USB, HDLC.

4.1.3.2



NRZ-коды

4.1.3.3

Закодируйте байт 10100110 кодами NRZ Level Inverted Unipolar, NRZ Space Bipolar.

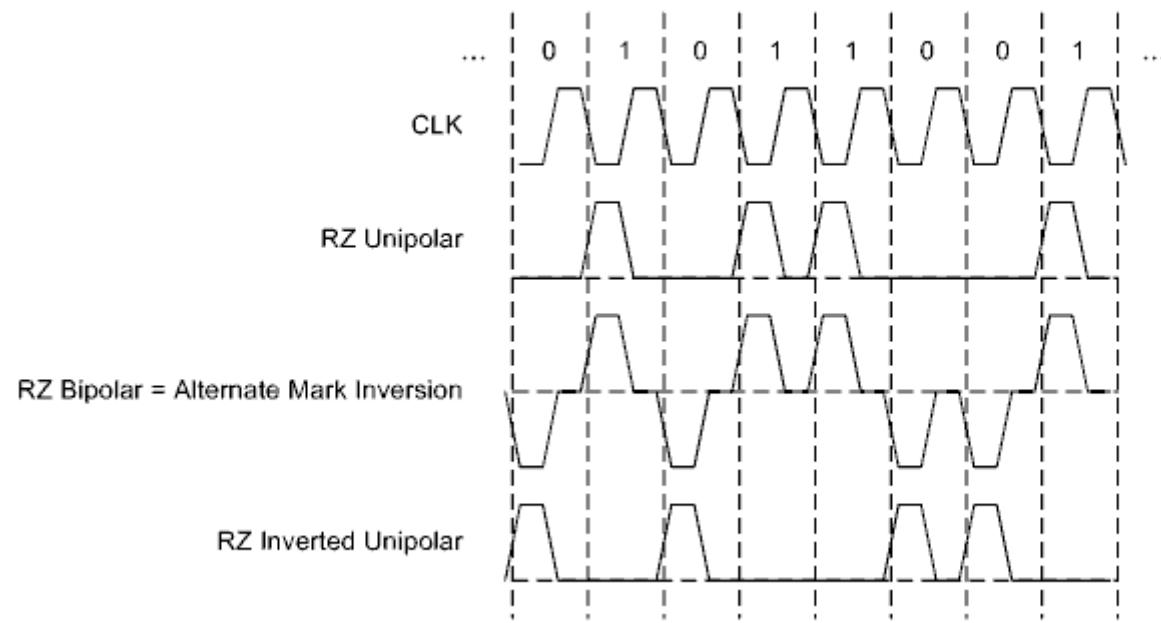
4.1.4.1

RZ-коды так же выражаются в изменении уровней между тактами, но на половине каждого такта всегда происходит возврат к нулю (земле).

Двухполярные RZ-коды обладают свойством самосинхронизации.

Пример технологии с применением RZ-кода: IrDA.

4.1.4.2



RZ-коды

4.1.4.3

Закодируйте байт 10100110 кодом RZ Inverted Bipolar.

4.1.5.1а

Манчестерские коды выражаются в переходах между уровнями во время тактов, поэтому их иногда называют фазовыми кодами.

Есть два «равноправных» варианта собственно манчестерского кода. Ноль во входной последовательности заменяется на переход от единицы к нулю, а единица заменяется на переход от нуля к единице. Либо наоборот.

Манчестерские коды обладают свойством самосинхронизации.

4.1.5.1b

Еще несколько кодов близки к манчестерскому.

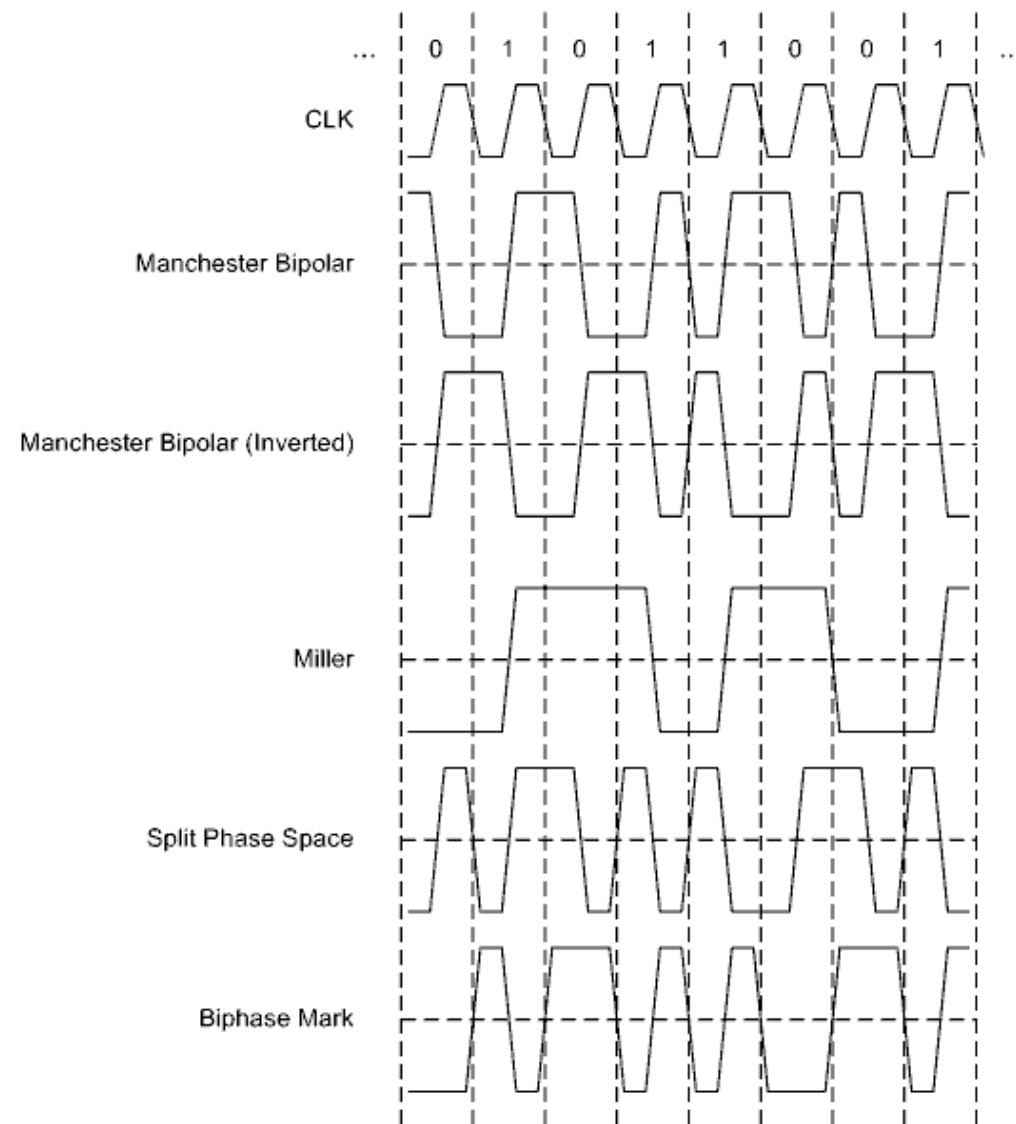
Согласно коду Миллера (Miller), ноль соответствует отсутствию перехода во время такта, единица соответствует переходу во время такта, плюс между двумя нулями всегда выполняется смена уровня.

Согласно коду Split Phase учитывается направление предыдущего перехода. При space-варианте ноль соответствует переходу во время такта в направлении, противоположном направлению предыдущего перехода, единица соответствуют переходу во время такта в направлении, совпадающем с направлением предыдущего перехода. При mark-варианте «роли» нулей и единиц из входной последовательности инвертируются.

Согласно коду Biphasе, кроме возможных переходов во время тактов, всегда выполняется смена уровня между тактами. При space-варианте ноль соответствуют переходу во время такта, единица соответствуют отсутствию перехода во время такта. При mark-варианте «роли» нулей и единиц из входной последовательности инвертируются.

Примеры технологий с применением манчестерских кодов: Ethernet, Token Ring, некоторые IR-технологии.

4.1.5.2



Манчестерские коды

4.1.5.3

Закодируйте байт 10100110 кодами Manchester Bipolar, Split Phase (Mark), Biphasе Space.

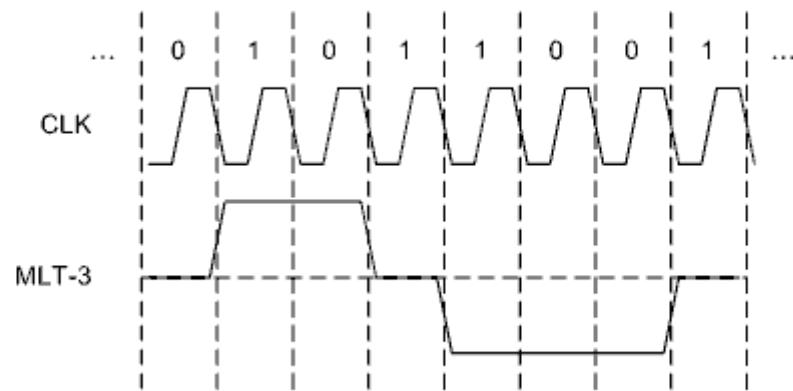
4.1.6.1

MLT-коды выражаются в переключении между несколькими уровнями между тактами.

Например, код MLT-3 имеет три уровня: -1, 0, +1. Кодирование может начинаться с нуля, ноль в исходной последовательности кодируется сохранением текущего уровня, а единица -- переходом к соседнему уровню (с сохранением направления, если это возможно).

Примеры технологий с применением MLT-кодов: Fast Ethernet, FDDI.

4.1.6.2



MLT-коды

4.1.6.3

Закодируйте байт 10100110 кодом MLT-3.

4.1.6.4

Что можно достичь увеличивая число уровней при кодировании?

4.1.7.1

Блочные коды выражаются в замене блоков битов из входной последовательности на бо'льшие (как правило) по размеру блоки битов в выходной последовательности.

Блочные коды могут комбинироваться с вышеперечисленными кодами.

В связи с избыточностью блочных кодов, во многих из них предусмотрены контрольные последовательности, которые, по сути, являются управляющими символами.

Первым примером может служить код 4b/5b, применяемый в Fast Ethernet и CDDI.

4.1.7.2

4b	5b
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

Основная таблица кода 4b/5b

4.1.7.3

Более сложным примером может служить код 8b/10b, применяемый в оптических вариантах Gigabit Ethernet.

Биты входного блока **обозначают** как $ABCDEFGH$ -- от младшего к старшему, выходного $abcdefghijkl$ -- так же от младшего к старшему.

Входной блок разбивается на два подблока: x из пяти битов и y из трех битов. Поэтому выходной код представляет собой конкатенацию двух кодов: 5b/6b и 3b/4b.

Кроме собственно блоков данных D , имеются контрольные блоки K , которые **кодируют** альтернативно.

Таким образом, входной блок **обозначают** как $Dx.y$ либо $Kx.y$.

Наконец, в код 8b/10b заложена гибкая система уравнивания количества нулей и количества единиц, заключающаяся в динамическом выборе блока для замены (одного из двух) исходя из текущего значения так называемого RD (Running Disparity). Предусмотрено два значение RD: -1 и +1. При выборе текущего значения RD учитывается предыдущее значение RD и соотношение нулей и единиц во входном блоке (плюс есть исключения).

4.1.7.4a

5b <i>EDCBA</i>		6b <i>abcdei</i>		5b <i>EDCBA</i>		6b <i>abcdei</i>	
<i>D</i>	RD = -1	<i>D</i>	RD = +1	<i>D</i>	RD = -1	<i>D</i>	RD = +1
00000	100111	011000		10000	011011	100100	
00001	011101	100010		10001		100011	
00010	101101	010010		10010		010011	
00011		110001		10011		110010	
00100	110101	001010		10100		001011	
00101		101001		10101		101010	
00110		011001		10110		011010	
00111	111000	000111		10111	111010	000101	
01000	111001	000110		11000	110011	001100	
01001		100101		11001		100110	
01010		010101		11010		010110	
01011		110100		11011	110110	001001	
01100		001101		11100		001110	
01101		101100		11101	101110	010001	
01110		011100		11110	011110	100001	
01111	010111	101000		11111	101011	010100	

Некоторые таблицы кода 8b/10b

4.1.7.4b

3b	4b	
HGF	<i>fghj</i>	
D	RD = -1	RD = +1
000	1011	0100
001	1001	
010	0101	
011	1100	0011
100	1101	0010
101	1010	
110	0110	
111	1110	0001
111	0111	1000

Некоторые таблицы кода 8b/10b

4.1.8.1

Линейные коды, применяемые в оптических каналах имеют особенности в сравнении с кодами для проводниковых каналов.

Примеры: TS-FO (Three of Six -- Fiber Optical), RZ carrier-suppressed, RZ alternate-phase.

ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

4.2.1.1

Очевидно, что неправильная интерпретация принятых данных чревата непредсказуемыми последствиями.

Серьезное изучение помехоустойчивого кодирования предполагает «погружение» в математику. При изучении же данной дисциплины больше важны прикладные аспекты, но без некоторых алгебраических основ не обойтись.

В теории помехоустойчивого кодирования очень важное место занимают поля Галуа, но чтобы к ним «подойти» нужно сделать ряд шагов.

4.2.1.2

Как в математике **задают** множество?

Приведите примеры множеств.

4.2.1.3

Некоторую операцию $*$ называют бинарной если после ее применения к двум любым элементам a и b некоторого множества получают элемент c , принадлежащий тому же множеству: $a * b = c$.

А соответствующее непустое множество S называют замкнутым относительно бинарной операции $*$.

Элемент e множества называют нейтральным если, после бинарной операции над этим элементом и некоторым другим, другой участвовавший в операции элемент не изменяется: $a * e = a$.

Два элемента множества называют обратными (относительно друг друга) если в результате бинарной операции над ними получают нейтральный элемент: $a * b = e$.

4.2.1.4

Множество G называют группой если для него определена бинарная операция $*$ и:

1. Операция $*$ является ассоциативной: $(a * b) * c = a * (b * c)$ -- соответствует умножению.
2. Существует нейтральный элемент -- соответствует единице.
3. Имеется унарная операция, позволяющая получить обратный элементу a элемент -- соответствует a^{-1} .

Группу называют абелевой если операция $*$ коммутативна: $a * b = b * a$.

Если для группы определена операция умножения ($a * b = ab$), то группу называют мультипликативной.

Мультипликативную группу называют циклической если в ней существует такой элемент, что все остальные элементы являются степенями этого элемента: $b = a^k$. А сам элемент a называют образующим группу.

4.2.1.5

Классом вычетов по модулю n , принадлежащему множеству натуральных чисел \mathbb{N} , называют подмножество элементов из множества целых чисел \mathbb{Z} , имеющих одинаковый остаток от деления на n .

[a] -- класс вычетов, одним из элементов которого является a .

Группу, образованную множеством классов вычетов по модулю n , называют группой классов вычетов по модулю n .

4.2.1.6

Приведите пример других чисел из класса вычетов по модулю 100, в котором содержится число 205.

4.2.1.7

Группу называют конечной если группа состоит из конечного числа элементов.

Число элементов $|G|$ конечной группы называют ее порядком.

4.2.1.8

Два целых числа сравнимы (эквивалентно равны) по модулю натурального числа n если при делении на n они дают одинаковые остатки: $a \equiv b \pmod{n}$.

4.2.1.9

Отображение $f: G \rightarrow H$ группы G в группу H называют гомоморфным если оно сохраняет операцию группы G . Отображение изоморфно если оно взаимно однозначно.

Отображение $f: G \rightarrow G$ называют эпиморфным, изоморфное отображение $f: G \rightarrow G$ называют автоморфным.

4.2.1.10

Множество R называют кольцом если для множества определены две бинарные операции $\#$ и $*$ такие что:

1. Множество R является абелевой группой относительно операции $\#$ -- соответствует сложению.
2. Операция $*$ является ассоциативной.
3. Выполняется закон дистрибутивности: $a * (b \# c) = a * b \# a * c$.

Если для группы определена операция сложения ($a \# b = a + b$), то группу называют аддитивной. Единичный элемент аддитивной группы соответствует нулю. Обратный элементу a элемент аддитивной группы соответствует $-a$.

На операцию $*$ можно накладывать дополнительные ограничения. Если в кольце присутствует единица, то кольцо называют кольцом с единицей.

При выполнении закона коммутативности кольцо называют коммутативным.

Коммутативное кольцо называют целостным если его единица не равна нулю и $a * b = 0$ только при $a = 0$ или $b = 0$.

4.2.1.11

Кольцо называют телом если кроме нуля в кольце существуют другие элементы и эти элементы образуют группу относительно операции *.

Наконец, коммутативное тело F называют полем.

Подгруппой, подкольцом, подполем называют подмножества сохраняющие соответствующие свойства.

Поле, не содержащее подполей, называют простым. Простым будет поле, порядок которого равен простому числу.

4.2.1.12

Подкольцо I кольца R называют его идеалом (двухсторонним идеалом) если для любой пары элементов a из I и r из R их произведение принадлежит I .

Подкольцо R/I классов вычетов по модулю идеала I из кольца R называют факторкольцом кольца R по идеалу I .

Наименьшее из натуральных чисел n , такое что для любого элемента r из кольца R выполняется равенство $n * r = 0$, называют характеристикой кольца R .

4.2.1.13

Согласно теореме, каждое конечное целостное кольцо образует поле.

Согласно другой теореме, характеристикой конечного поля является простое число.

Поле $GF(p)$ из целых чисел $0, 1 \dots p - 1$, порожденное в результате отображения $f: \mathbb{Z}/p \rightarrow GF(p)$, где \mathbb{Z}/p -- факторкольцо множества целых чисел, в котором роль идеала играет простое число p , и $f([a]) = a$, называют полем Галуа (Galois field) порядка p .

При вычислениях с элементами поля Галуа используют целочисленную арифметику с приведением по соответствующему модулю.

4.2.1.14

Задача у доски.

Из каких элементов будет состоять поле $GF(5)$?

Задайте операции сложения и умножения для поля $GF(5)$?

4.2.1.15

Одно и то же число можно записать самыми разными способами. Число можно рассматривать и как значение полинома.

Полиномом (многочленом) одной переменной называют выражение:

$$f(x) = \sum_{i=0}^n a_i x^i = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0 .$$

В случае *полинома над кольцом*, коэффициенты a_i соответствуют элементам кольца, а переменная x кольцу не принадлежит.

Два полинома $f(x)$ и $g(x)$ называют равными если равны их степени и равны коэффициенты при одинаковых степенях переменной.

Если старший коэффициент полинома равен единице, то полином называют приведенным.

4.2.1.16

Что такое НОД и НОК?

Для чего **применяют** алгоритм Евклида?

4.2.1.17

Тривиальными делителями полинома называют сам полином и полином равный 1.

Полином над полем F называют неприводимым над этим полем если полином допускает только тривиальное разложение, то есть у полинома нет нетривиальных делителей из F (с точностью до домножения на ненулевую константу).

4.2.1.18

Запишите все неприводимые полиномы третьей степени.

Что такое взаимно простые числа?

4.2.1.19

Что такое корень полинома?

4.2.1.20

Для практического применения полей Галуа в компьютерных системах необходимо перейти от скалярного представления к векторному.

Расширенное поле Галуа $GF(p^n)$ можно рассматривать как векторное пространство, где простое число p является характеристикой поля и соответствует количеству состояний разряда вектора, а n является степенью поля над его простым подполем и соответствует количеству разрядов вектора.

Поскольку в обычных компьютерных системах разряды регистров бинарные, то наибольший интерес представляют поля $GF(2^n)$.

4.2.1.21а

Сложение бинарных векторов (совпадает с вычитанием) проблему не представляет и соответствует поразрядной операции xor.

А вот с умножением и делением дела обстоят значительно сложнее.

Скалярное произведение не подходит, так как его результат может «выйти» за пределы поля.

Векторное произведение определено только для трехразрядных векторов.

Полиномиальное представление так же с ходу не решает проблему, так как произведение полиномов опять же «выводит» за пределы поля.

Для обеспечения конечности поля Галуа, полученный в результате произведения полином нужно привести. Это достигают путем деления на некий выбранный полином степени n . Ясно, что выбирать можно разные полиномы. Выбор другого полинома приведет к другим результатам умножения и, соответственно, к другому полю $GF(p^n)$.

Выбранный для построения поля Галуа полином называют порождающим (образующим).

4.2.1.21b

Деление векторов в математике не известно.

После перехода на язык полиномов, опять же для обеспечения конечности поля Галуа, деление всегда должно быть безостаточным. Деление можно представить как умножение полинома-делимого на полином, обратный делителю. При этом для достижения цели на основании математических выкладок, необходимо ввести еще одно ограничение: порождающий полином должен быть неприводимым по модулю p (например, если $p = 2$ и $n = 4$, то полином $x^4 + 1$ (число 17) не подходит, так как $x^4 + 1 \equiv (x^2 + 1)^2 \pmod{2}$).

Возведение в степень обладает цикличностью.

4.2.1.22

Задача у доски.

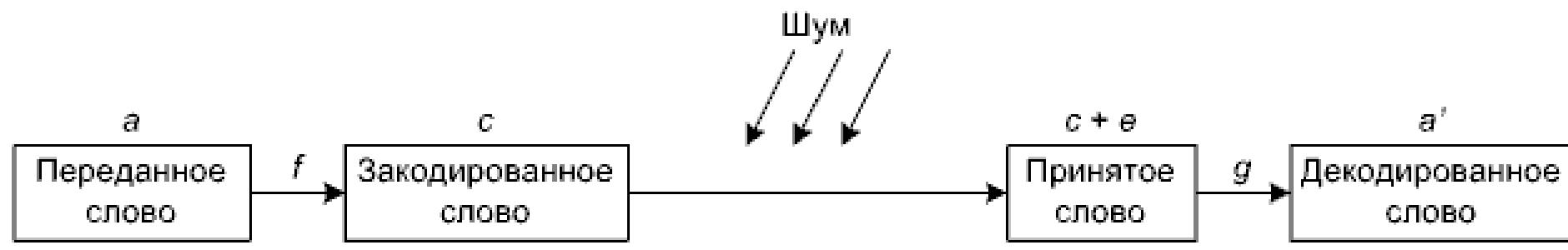
Как будут выглядеть элементы поля $GF(2^3)$ с порождающим полиномом $x^3 + x^2 + 1$?

Определите операции сложения и умножения.

4.2.2.1

Считается, что начало помехоустойчивому кодированию положила теорема Шеннона, утверждающая что любой дискретный канал связи имеет конечную пропускную способность и этот канал может быть задействован для передачи информации со сколь угодно большой степенью достоверности, не смотря на наличие помех.

4.2.2.2a



Модель канала связи

4.2.2.2b

Передаваемое сообщение разбивается на блоки фиксированного размера a из k битов $a_1, a_2 \dots a_k$.

Кодер выполняет функцию f , называемую схемой кодирования, и тем самым преобразует вектор a в вектор c из $n > k$ битов $c_1, c_2 \dots c_n$, называемый **кодовым словом**.

В процессе пересылки кодового слова по каналу связи на него накладывается вектор ошибок e , в котором единичные биты соответствуют искажениям.

После применения декодером схемы декодирования g получается вектор a' , в идеале совпадающий с исходным вектором a .

4.2.2.3

Подобная схема кодирования является избыточной. На практике всегда **ищут** компромисс между степенью обеспечения достоверности при передаче и вычислительной сложностью кодов (что в первую очередь отражается на скорости декодирования).

В КС множество кодовых слов получается из множества исходных слов как отображение из конечного поля $GF(2^k)$ в конечное поле $GF(2^n)$.

При более простых схемах кодирования, в кодовом слове сначала располагаются биты входного сообщения, называемые *информационными*, а за ними дополнительные биты, называемые *проверочными*: $a_1, a_2 \dots a_k, c_{k+1}, c_{k+2} \dots c_n$.

В более сложных случаях проверочные биты чередуются с информационными.

4.2.2.4a

Схему кодирования удобно представлять в матричном виде.

Схема кодирования:

$$f: GF(2^3) \rightarrow GF(2^6) = a_1, a_2, a_3 \rightarrow a_1, a_2, a_3, c_4, c_5, c_6$$

Проверочные уравнения:

$$\begin{aligned}c_4 &= a_1 \wedge a_2 \\c_5 &= a_2 \wedge a_3 \\c_6 &= a_1 \wedge a_3\end{aligned}$$

Переход к матричному представлению:

$$\begin{bmatrix} c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Проверочные уравнения по-другому:

$$\begin{aligned}a_1 \wedge a_2 \wedge c_4 &= 0 \\a_2 \wedge a_3 \wedge c_5 &= 0 \\a_1 \wedge a_3 \wedge c_6 &= 0\end{aligned}$$

В матричном виде (T означает транспонирование):

$$H c^T = 0$$

4.2.2.4b

Видно, что проверочные уравнения образуют систему линейных уравнений. Следовательно, отображение f (схема кодирования) является линейным.

4.2.2.5

Если H -- матрица размером $(n - k) \times n$ ранга $n - k$ и $H c^T = 0$, то множество всех n -разрядных векторов, входящих в поле $GF(2^n)$ (в общем случае $GF(p^n)$), называют линейным (n, k) -кодом (в математическом смысле) длины n и размерности k . А матрицу H называют проверочной.

Линейный код по-другому называют групповым, так как множество кодовых слов можно рассматривать как подгруппу в отношении поля $GF(2^n)$.

Линейный код называют систематическим (разделенным) если расположение проверочных битов известно (не важно где они находятся), то есть если $H = [A \quad I_{n-k}]$, где A -- матрица размером $(n - k) \times k$, а I_{n-k} -- единичная матрица ранга $n - k$.

4.2.2.6

Матрицу $G = [I_k \quad -A^T]$ размером $k \times n$ называют **кодирующей** (порождающей) матрицей систематического кода.

Кодирующая и проверочная матрицы связаны следующим образом:
 $GH^T = 0$.

4.2.2.7

Самыми примитивными из линейных кодов являются подсчет контрольной суммы и дублирование информационных символов.

4.2.2.8

Перед выбором того либо иного помехоустойчивого кода всегда нужно определиться, что требуется от кода. Если перефразировать, то нужно ответить на два вопроса:

1. Сколько бинарных ошибок код должен обнаруживать.
2. Сколько бинарных ошибок код должен исправлять.

Исправлять ошибки значительно сложнее, чем обнаруживать. Применительно ко многим кодам, исправление ошибки подразумевает нахождение ее позиции.

4.2.2.9

В общем случае ошибки носят случайный характер. Множественные ошибки могут быть взаимозависимыми, то есть образовывать *модули ошибок*. Если ошибки расположены рядом, то они образуют *пакет ошибок* (частный случай модуля).

4.2.2.10

Число координат (позиций), которыми два вектора x и y различаются **называют расстоянием Хэмминга** -- $d(x,y)$.

Число ненулевых позиций вектора x **называют весом Хэмминга** -- $w(x)$.

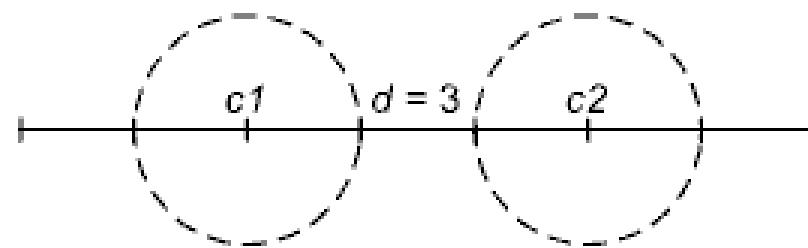
Видно, что расстояние Хэмминга показывает количество возникших ошибок.

4.2.2.11

Для увеличения корректирующей способности кода следует стремиться увеличивать расстояния между кодовыми словами. При этом минимальное расстояние d_{min} называют кодовым и оно является очень важной характеристикой помехоустойчивого кода.

Согласно теореме, для того чтобы линейный код исправлял t ошибок должно выполняться условие: $d_{min} \geq 2t + 1$.

Для того, чтобы линейный код обнаруживал t ошибок должно выполняться условие: $d_{min} \geq t + 1$.



Для того чтобы линейный код имел $d_{min} \geq s + 1$, необходимо и достаточно, чтобы любые s столбцов его проверочной матрицы были линейно независимы.

4.2.2.12

Способность того или иного кода сохранять свои характеристики зависит и от количественного соотношения информационных и проверочных символов. В теории помехоустойчивого кодирования определяют так называемые верхние и нижние границы кодов.

4.2.2.13

Существуют три основных способа декодирования:

1. По минимуму расстояния.
2. По синдрому (по лидеру смежного класса).
3. Мажоритарное:
 - с разделенными проверками;
 - с λ -проверками;
 - с квазиразделенными проверками.

4.2.3.1

За достаточно длительную историю развития прикладной теории кодирования, как науки, было придумано очень много помехоустойчивых кодов.

Основные группы помехоустойчивых кодов:

1. Линейные коды, в том числе: коды Хэмминга, циклические коды, БЧХ-коды (коды Боуза-Чоудхури-Хоквингема), РМ-коды (коды Рида-Маллера), итеративные коды, коды на основе матриц Адамара, симплексные коды и некоторые другие.
2. Коды для контроля модульных и пакетных ошибок, в том числе: РС-коды (коды Рида-Соломона), низкоплотные модульные коды, векторные модульные коды, итеративные модульные коды и некоторые другие.
3. Сверточные коды.
4. Арифметические коды.
5. Низкоскоростные коды, в том числе: коды максимальной длины, нелинейные коды, D-коды и некоторые другие.

4.2.3.2

Для изучения в рамках данной дисциплины **выбраны** два кода:

1. Код Хэмминга -- Hamming code.
2. Циклический код -- CRC (Cyclic Redundancy Code).

4.2.4.1

Бинарным кодом Хэмминга **называют** код длины $n = 2m - 1$, $m \geq 2$ с проверочной матрицей H размером $m \times (2m - 1)$, в которой столбцы соответствуют записи $1, 2 \dots 2^m - 1$ в двоичной системе счисления.

Код Хэмминга позволяет исправлять одиночную ошибку и обнаруживать множественные ошибки.

4.2.4.2

Задача у доски.

Запишите проверочную матрицу кода Хэмминга (7,4).

4.2.5.1

Циклические коды являются особо выделяемой подгруппой линейных кодов.

Циклическим кодом называют линейный код, удовлетворяющий дополнительному условию: если вектор $a_0, a_1 \dots a_{n-1}$ является кодовым словом, то и его циклический сдвиг $a_{n-1}, a_0 \dots a_{n-2}$ так же является кодовым словом.

Циклический код позволяет исправлять одну и более ошибок и обнаруживать множественные ошибки (зависит от параметров).

4.2.5.2

Базовая идея циклического кодирования состоит в том, чтобы в качестве проверочных битов передавать остаток от деления информационных битов на некоторое выбранное число.

После приема снова выполняется деление уже возможно искаженных информационных битов на то же самое число и сравниваются остатки.

Если остатки совпадают, то данные с определенной вероятностью приняты без ошибок.

4.2.5.3

На практике же деление выполняется по правилам арифметики полей Галуа, то есть без учета переносов.

Информационные биты, то есть делимое, соответствуют информационному полиному.

Делитель соответствует порождающему (образующему) полиному.

Частное в процессе кодирования не используется и поэтому «отбрасывается».

Для того чтобы максимально разнообразить остатки в качестве порождающего полинома должен выбираться неприводимый полином.

4.2.5.4

Существуют два подхода к реализации циклического кода на стороне приемника:

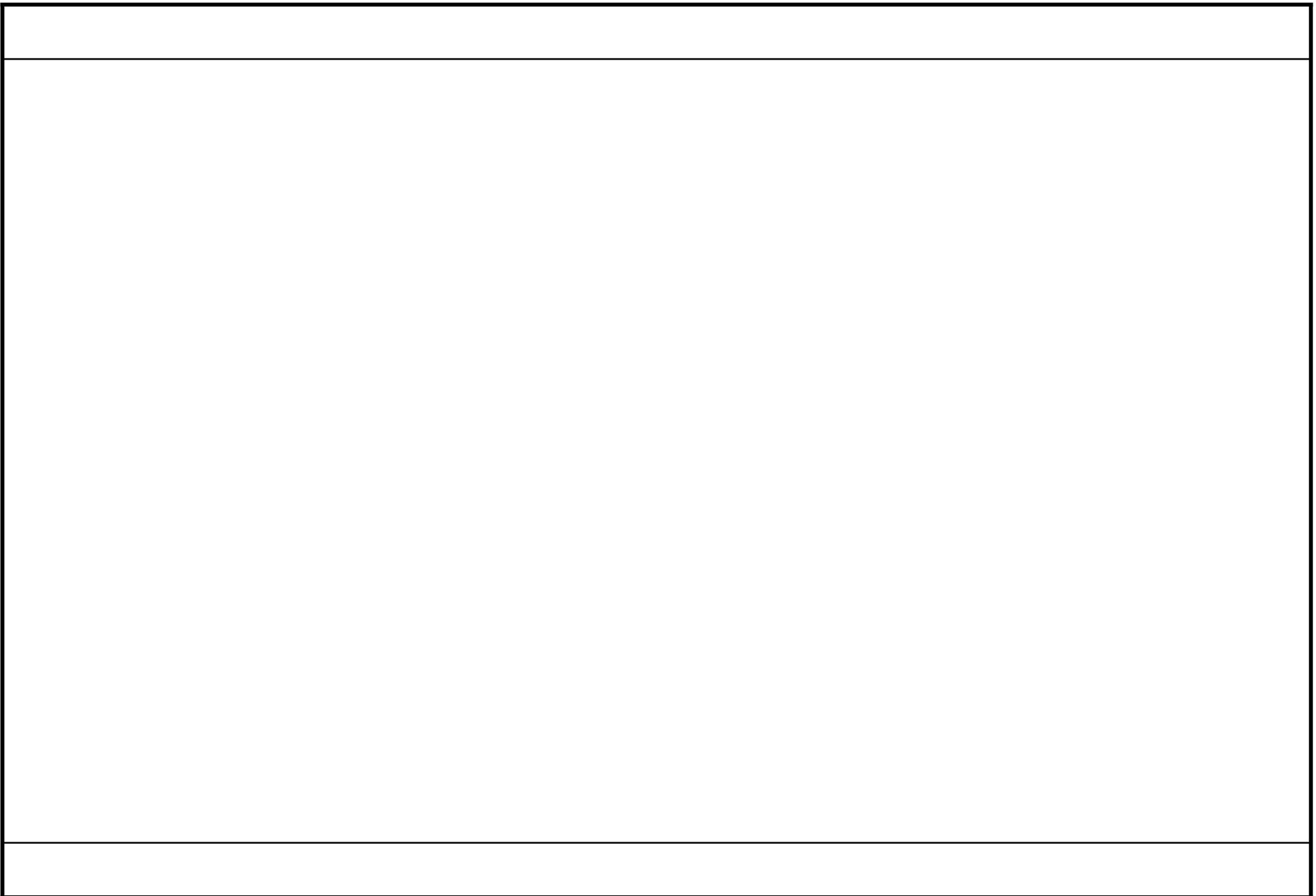
1. Согласно базовой идее, описанной выше.
2. На порождающий полином делится все принятое кодовое слово. Если ошибок не произошло, то остаток будет нулевым.

Оба подхода равноценны.

4.2.5.5

Задача у доски.

Сформируйте циклический код для информационного вектора 10100110
(требуется обнаружить и исправить одиночную ошибку).



ТОПОЛОГИИ КОМПЬЮТЕРНЫХ СЕТЕЙ

5.0.1.1

При разговоре о топологиях КС не обойтись без привязки к уровням модели OSI.

На физическом уровне оперируют с сигналами, поэтому концентрируются на отдельно взятом передатчике, отдельно взятом приемнике и последовательном канале, который их связывает. Для обеспечения модуляции этого вполне достаточно. При этом приемник активен всегда, а передатчик включается по мере надобности.

Конечно, передатчиков и приемников может быть много, что не может не накладывать определенный отпечаток на физические процессы, но топология, как таковая, при этом особого интереса не представляет.

5.0.1.2

С точки зрения направленности, последовательный канал может функционировать в одном из трех режимов:

1. *Симплексом* (simplex) -- передача данных по каналу возможна только в одном направлении.
2. *Полудуплексом* (semiduplex) -- данные могут передаваться в обоих направлениях, но в один момент времени возможна передача только в одном направлении.
3. *Полнодуплексом* (full duplex) -- данные могут передаваться в обоих направлениях одновременно.

Сейчас в КС доминируют полнодуплексные каналы.

5.0.2.1

Топология «возникает» на канальном уровне, когда речь идет об организации сегмента.

5.0.2.2

Последовательный канал может быть:

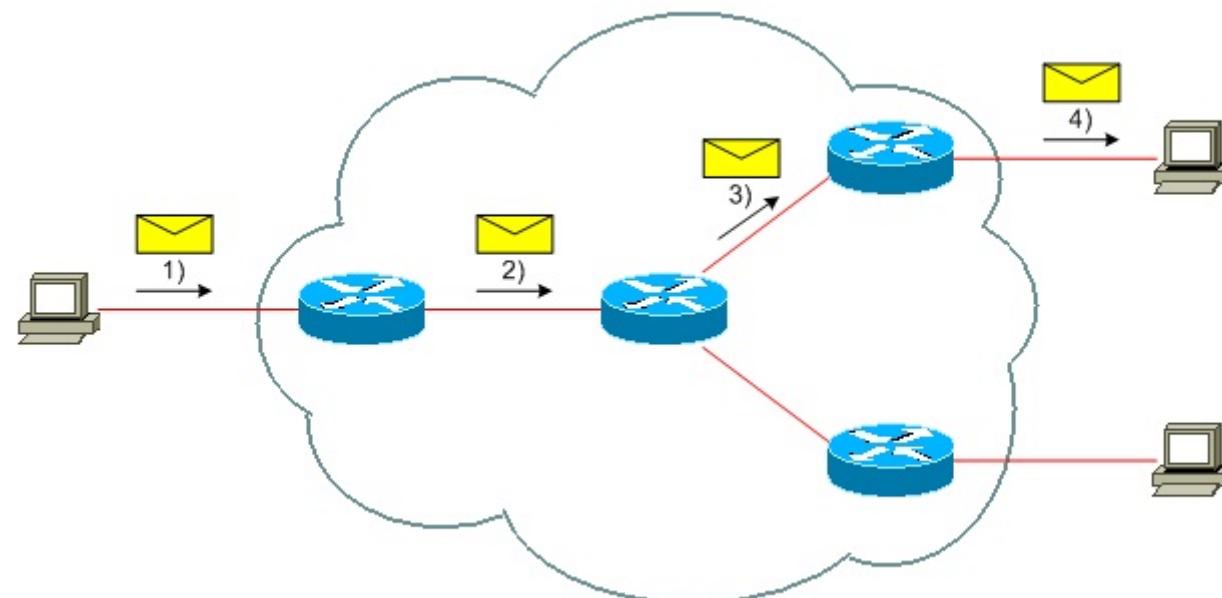
1. *Выделенным* (leased) -- зарезервирован за определенной парой станций-абонентов.
2. *Разделяемым* (shared) -- может использоваться несколькими станциями-абонентами.

Причем канал, который не может разделяться несколькими станциями-передатчиками одновременно, в отечественной литературе принято называть моноканалом. Во многих реализациях ситуация именно такая.

5.0.2.3а

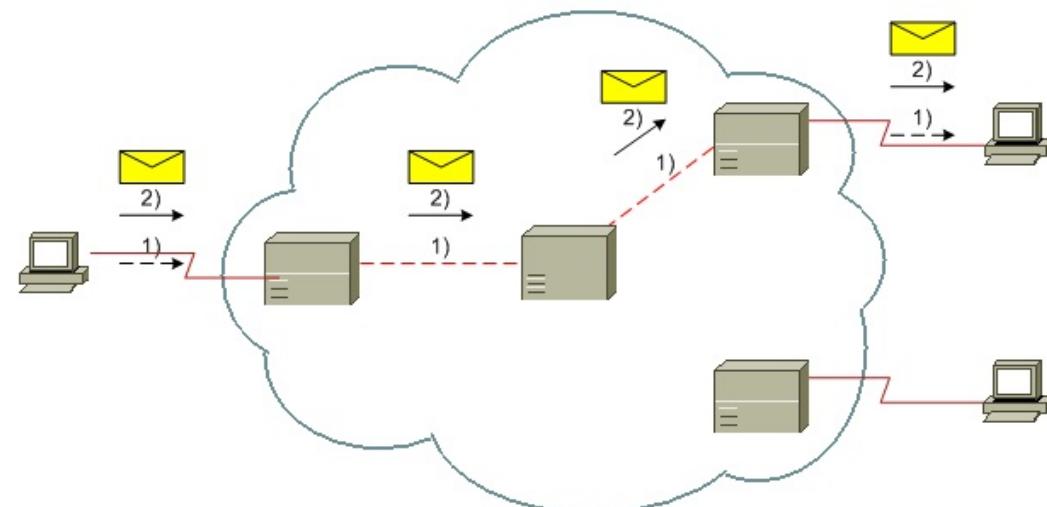
С точки зрения общей организации процесса пересылки данных, все СПД можно разделить на два фундаментальных типа:

1. *СПД с коммутацией пакетов* (packet-switched) -- в структуру пакетов включают адреса станций-абонентов; каждое устройство-посредник определяет дальнейший путь на основании анализа адресов назначения; каждый из пакетов в цепочке пересыпается независимо от остальных (следует учитывать, что в сложных СПД имеются альтернативные пути пересылки).



5.0.2.3b

2. СПД с коммутацией каналов (circuit-switched) -- адреса станций-абонентов в структуру пакетов (кадров) не включают; сначала, по запросу станции-передатчика, на основании запрашиваемого адреса, СПД «прокладывает» к вызываемой станции канал, называемый коммутируемым; каналы-звенья могут быть как выбранными «целиком» каналами (как правило между оконечными устройствами и устройствами-посредниками), так и выбранными подканалами каналов с частотным или времененным разделением (как правило между устройствами-посредниками); затем созданный канал используется для пересылки пакетов (кадров).



5.0.2.4

Прежде всего, топологии делят на два типа:

1. Point-to-point -- топология «точка к точке» -- связывает только две станции.

2. Multi-access (multipoint-to-multipoint) -- топология с множественным доступом -- связывает более двух станций.

Эти два типа позволяют организовывать двунаправленные каналы между любым требующимся количеством абонентов, поэтому их реализуют наиболее часто.

Применительно к односторонним каналам можно добавить еще два пункта:

+3. Point-to-multipoint -- иногда.

+4. Multipoint-to-point -- очень редко.

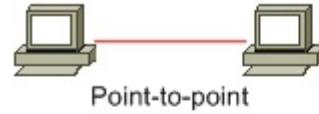
Менее двух станций в сегменте быть не может.

5.0.2.5

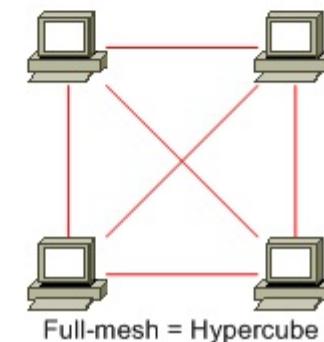
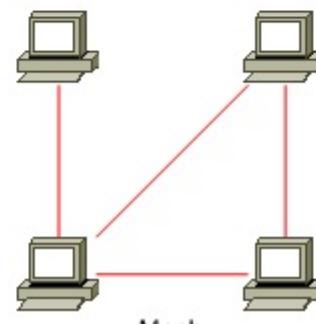
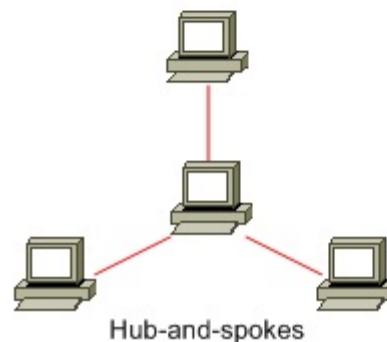
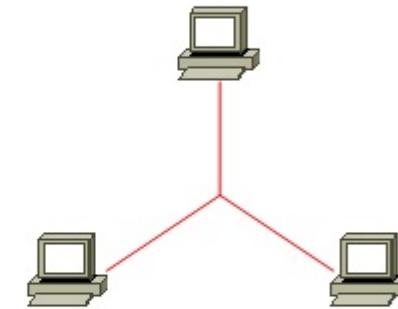
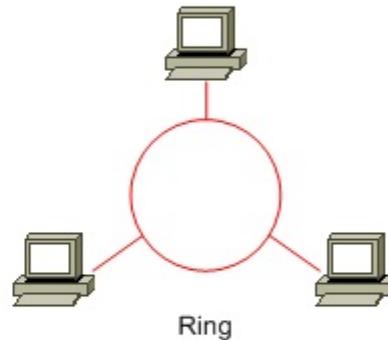
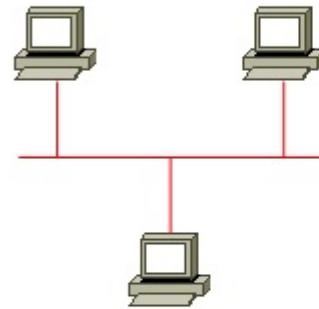
Попробуйте нарисовать несколько топологий сегментов КС с детализацией до станций и СрПД.

5.0.2.6

Point-to-point:



Multipoint:



Топологии КС с детализацией до станций и СрПД

5.0.2.7

В общем случае, направленность каналов может «накладываться» на топологии по-разному. Например, кольцо может быть односторонним и двунаправленным.

Сегмент может иметь и *гибридную топологию* (hybrid topology).

5.0.2.8

Чем физическая топология сегмента отличается от логической?

5.0.2.9

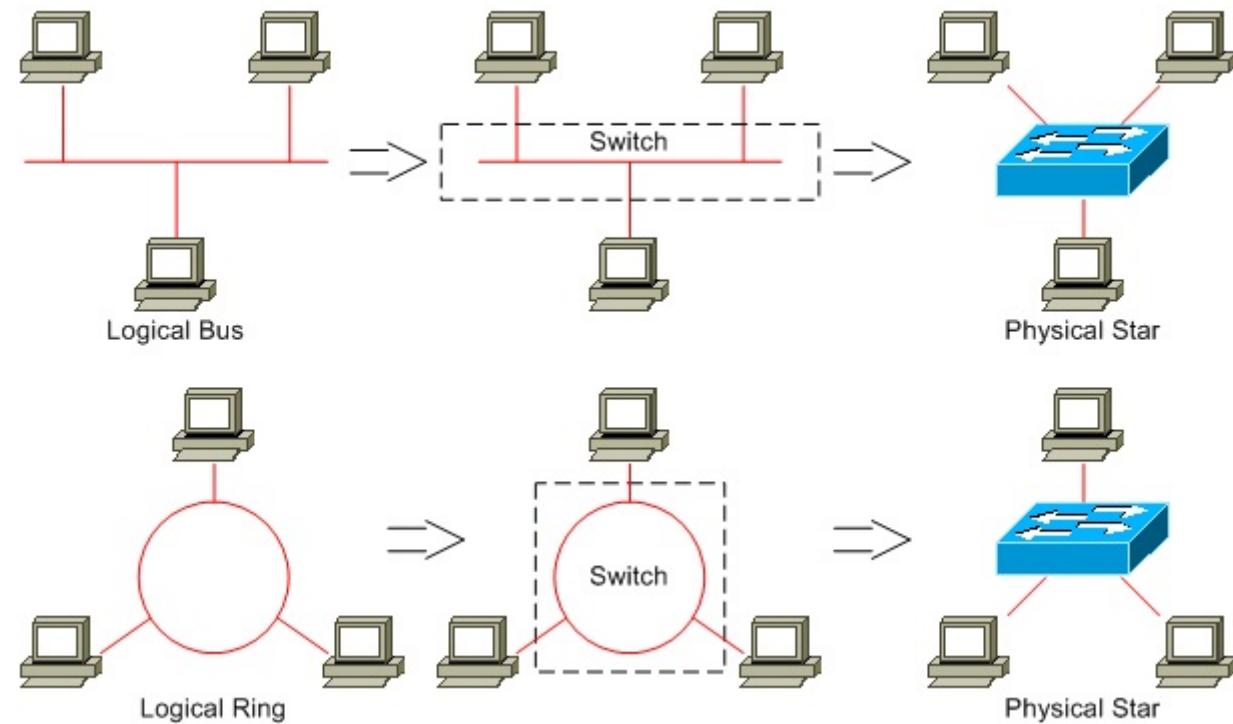
Если топологически классифицировать аппаратные технологии (охватывающие два нижних уровня модели OSI), то есть еще два ракурса:

1. *Физическая топология* (physical topology) -- отражает физические связи между устройствами.

2. *Логическая топология* (logical topology) -- отражает логические связи между устройствами.

Часто логическая топология не совпадает с физической.

5.0.2.10



Примеры соответствий между физическими и логическими топологиями

5.0.2.11

Характерными топологиями ЛКС являются:

1. Шина (bus).
2. Кольцо (ring).
- +3. Звезда (star).

5.0.2.12

Характерными топологиями ГКС являются:

1. Сеть (произвольно связанная) (mesh).
- +2. Ступица со спицами (hub-and-spokes).
- +3. Полносвязная сеть (full-mesh).

Характерной RAS-топологией является point-to-point.

Можно сказать, что для ГКС-технологий существует только одна типичная топология (произвольно связанная сеть), остальные можно рассматривать как ее частные случаи.

Для RAS-технологий существует только одна типичная топология.

На начальных этапах изучения, Cisco не отделяет RAS от ГКС.

5.0.3.1

Сегменты соединяют произвольным образом, поэтому на сетевом уровне уместно говорить о топологии с произвольными связями, хотя топологию в отношении третьего уровня упоминают весьма редко.

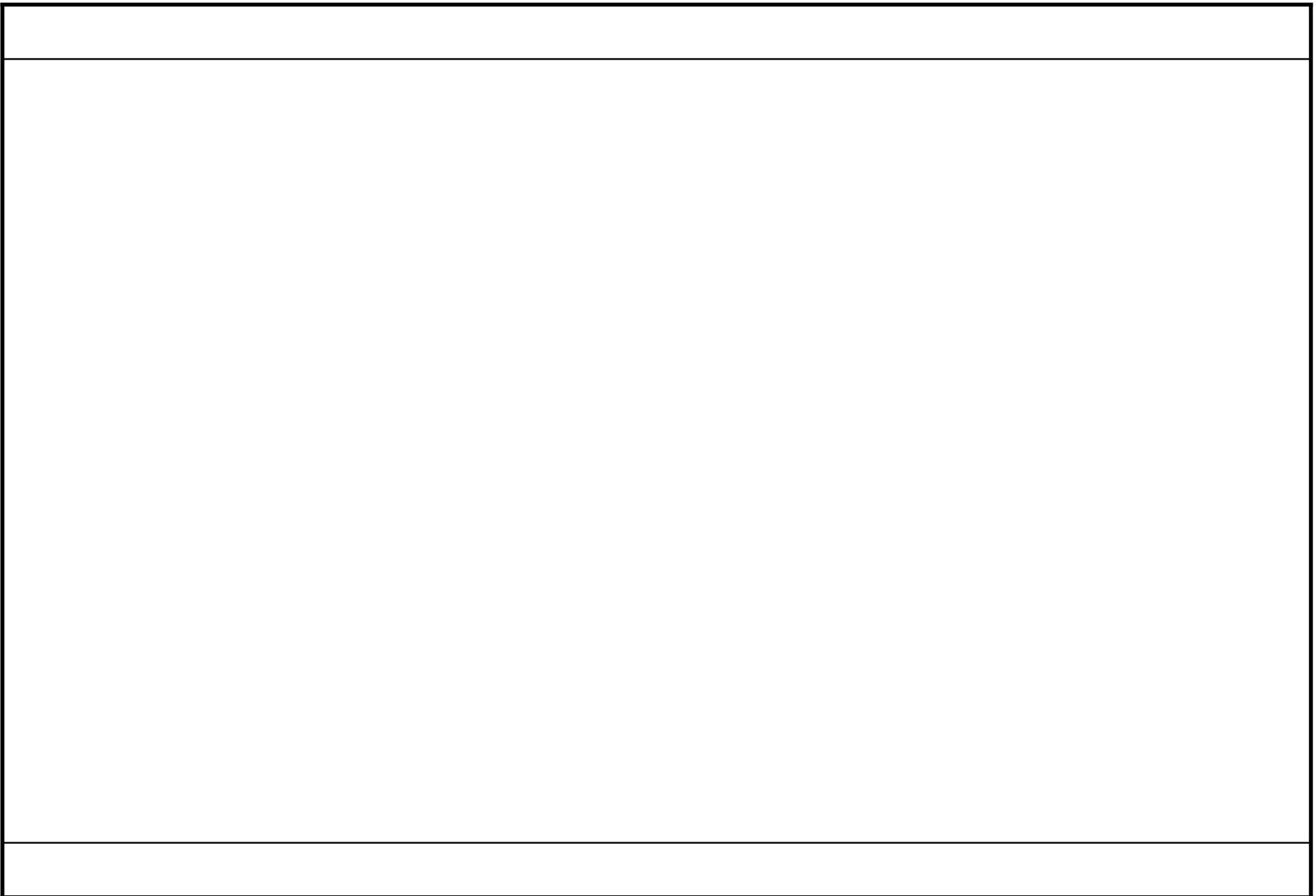
Протоколы сетевого уровня обычно разрабатывают топологически независимыми.

5.0.4.1

Начиная с транспортного уровня, топологии применимы к связям между программами, производящими и потребляющими сетевые услуги, поэтому могут быть только логическими.

Здесь характерными топологиями являются:

1. Point-to-point -- при двунаправленной передаче.
2. Point-to-multipoint -- при однонаправленной передаче.



СЛУЧАЙНЫЕ МЕТОДЫ ДОСТУПА К МОНОКАНАЛУ

6.0.1.1

Различные алгоритмы доступа к моноканалу разрабатывают по причине необходимости разрешения конфликтов между станциями при взаимодействии посредством разделяемой СрПД.

6.0.1.2

В первую очередь алгоритмы затрагивают передатчики, то есть активные компоненты системы. Проблема заключается в «столкновениях» конкурирующих передатчиков.

Пассивные по своей природе приемники априори конфликтовать не могут. Хотя количество приемников всегда ограничено, так как передатчики имеют конечную нагрузочную способность.

Если находящиеся в равных условиях два либо более передатчиков одновременно выдают сигналы в СрПД (например, устанавливают соответствующие уровни напряжения), то возникает противоречие. Таковое единовременно неразрешимое противоречие принято называть коллизией (collision).

6.0.1.3

Коллизия может быть как логической (информационный конфликт) так и физической (несовместимые физические процессы).

Обычно коллизия возникает при попытках установить противоположные логические уровни.

Кроме всего прочего, физическая коллизия чревата выходом из строя передатчиков, даже при попытках установить одинаковые логические уровни, так как многие среды не допускают наличие более чем одного активного усилителя сигнала без применения специальных схемотехнических решений.

Классическим способом защиты оборудования от коллизий является так называемая гальваническая развязка (трансформаторная либо оптронная).

При попытках установить разные уровни, как правило, наблюдают эффекты «зануления» и «заединчивания» -- в зависимости от особенностей элементной базы.

6.0.1.4

Ситуация с коллизией может затрагивать только станции, подключенные к одной СрПД, то есть сегмент компьютерной сети.

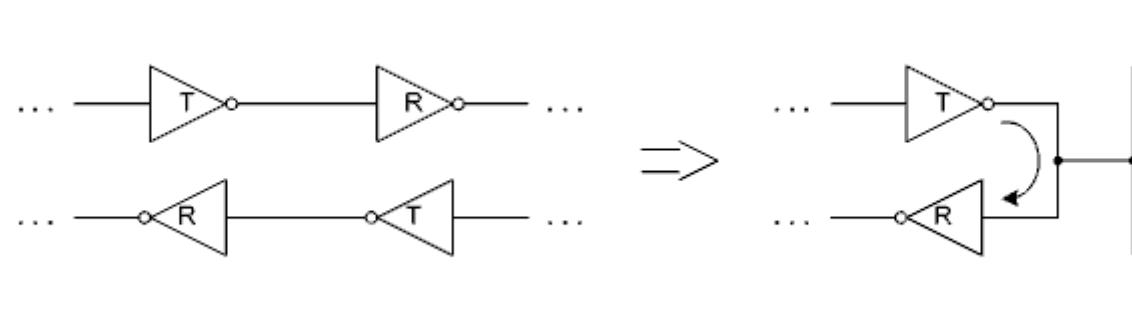
Сегмент, в котором возможно возникновение коллизий называют *доменом коллизий* (*collision domain*).

Понятие коллизии имеет отношение не только к сигналу, а и к пакету.

6.0.1.5а

Чтобы передатчик мог бороться с коллизиями он безусловно должен иметь возможность определять факт их наличия.

Борьба с коллизиями, по определению, актуальна применительно к многоточечным топологиям (если под точками понимать подключения), типичными представителями которых являются шинная топология и эфир. На практике, при переходе от двухточечной топологии к многоточечной цепи передатчика и приемника всегда совмещают.



Получить легко наращиваемую структуру по-другому невозможно.

6.0.1.5b

Ответ на вопрос о том как передатчик определяет наличие коллизии весьма удачно «заложен» в показанную схемотехнику. Можно легко заметить, что любая переданная передатчиком порция данных, например байт, тут же будет принята приемником. Достаточно просто сравнить (аппаратно или программно) байт до передачи с байтом после приема. Несовпадение свидетельствует о том, что была коллизия. Ну а если вдруг даже после коллизии данные совпали, то это «устраивает все стороны».

6.0.1.6

Физические свойства СрПД не позволяют мгновенно передавать сигналы. Следовательно и возникшая коллизия распространяется по сегменту с конечной скоростью.

Под окном коллизий (collision window) понимают временной интервал, в течение которого любая из станций гарантированно обнаруживает коллизию, равный удвоенному времени прохождения сигнала между двумя максимально удаленными станциями.

Без учета окна коллизий, влияющего на время постудержания сигнала, невозможно спроектировать работоспособный сегмент.

6.0.1.7

Почему окно коллизий равно удвоенному времени прохождения сигнала между двумя максимально удаленными станциями?

6.0.2.1

Существуют два основных подхода к проблеме коллизий:

1. Не допускать коллизии вообще, то есть пользоваться детерминированными методами доступа к моноканалу.

2. Допускать коллизии и каким-то образом выходить из них, что достижимо только использованием случайных методов доступа к моноканалу.

Во втором случае так же можно выделить два подхода:

1. Не обращать внимание на причины возникновения коллизий, а упор делать на способ выхода из них.

2. Пытаться предотвращать коллизии тем самым максимально снижая их количество, ну а если коллизии все-таки возникают, то «тяжело» выходить из них.

Таким образом, все методы доступа к моноканалу делят на:

1. Случайные (contention-based).
2. Детерминированные (controlled).

6.0.2.2

Все случайные методы основаны на использовании генератора случайных чисел (поэтому их так и называют), который позволяет делать случайные задержки при доступе к моноканалу, а значит и с определенной степенью вероятности избегать коллизии.

6.0.2.3

На эффективность случайных методов наиболее существенное влияние оказывают следующие факторы:

- количество взаимодействующих станций;
- инертность среды передачи данных;
- длина кадра;
- частота синхронизации.

6.0.2.4

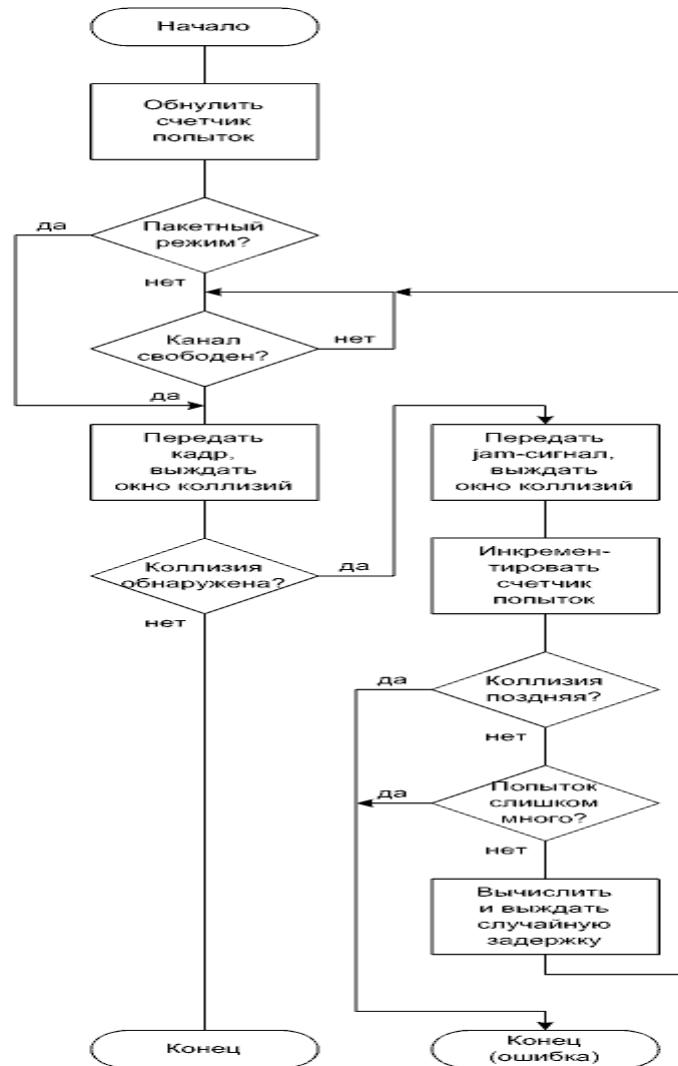
Назовите основные достоинства случайных методов доступа.

Назовите основные недостатки случайных методов доступа.

6.0.3.1

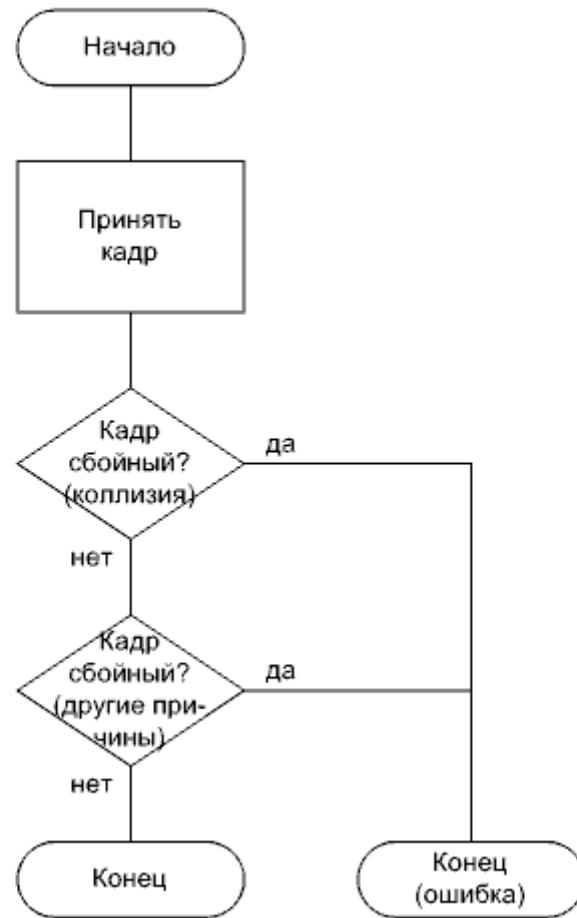
С точки зрения изучения случайных методов доступа к моноканалу наиболее наглядным примером является классический алгоритм CSMA/CD (Carrier Sense Multiple Access with Collision Detection) -- множественный доступ с прослушиванием несущей и обнаружением коллизий, описанный в стандарте Ethernet (IEEE 802.3).

6.0.3.2



Алгоритм CSMA/CD. Передача очередного кадра

6.0.3.3



Алгоритм CSMA/CD. Прием очередного кадра

6.0.3.4

Задержку перед началом очередной попытки передачи после коллизии (backoff) измеряют в так называемых слот-таймах, количество которых является случайным целым числом r :

$$0 \leq r \leq 2^k ,$$

где

$$k = \min(n, 10) ,$$

где n -- номер попытки.

После превышения счетчиком попыток некоторого порогового значения дальнейшие попытки считаются бесперспективными.

Значение n не может быть больше 16, а значение k не может быть больше 10.

6.0.3.5

Качество диспетчеризации при обработке коллизий по большому счету зависит от одного базового параметра.

Слот-тайм (slot time) является минимальной неделимой единицей времени при диспетчеризации. Слот-тайм подбирают с учетом многих других параметров. По крайней мере, он должен быть больше суммы удвоенного времени прохождения сигнала по сегменту и времени передачи јат-сигнала.

6.0.3.6

Назовите две причины, по которым нужен јат-сигнал.

6.0.3.7

В стандарт заложен механизм ускорения распределенного обнаружения коллизий, заключающийся в их «усилении».

Каждая обнаружившая коллизию станция передает специальный *jam-сигнал* некоторой длительности (значение стандартом не регламентировано).

Jam-сигнал выполняет две важные функции. Во-первых, является признаком возникновения коллизии, что позволяет другим станциям сразу «увидеть» коллизию (столкнувшиеся передатчики, выставившие jam-сигнал, и так знают о коллизии). Во-вторых, позволяет синхронизировать начала отсчетов случайных задержек.

6.0.3.8

7 B	1 B	6 B	6 B	2 B	46 – 1500 Bytes	4 B	?
Preamble	SFD	DA	SA	Length/ Type	Data	Pad	FCS Extensi on

Поля:

Preamble -- преамбула.

SFD (Start Frame Delimiter) -- разграничитель начала кадра.

DA (Destination Address) -- адрес назначения.

SA (Source Address) -- адрес источника.

Length/Type -- длина либо тип.

Data -- данные.

Pad -- наполнитель.

FCS (Frame Check Sequence) -- контрольная сумма.

Extension -- расширитель.

Формат кадра Ethernet

6.0.3.9

Предусмотрены полу duplexный и полнодуплексный режимы, «поведение» в которых несколько различается.

В качестве преамбулы выступают семь байтов со значением 10101010b, а в качестве SFD -- байт со значением 10101011b.

При сборке кадра учитываются ограничения на его длину. Ограничиваются не только максимальная длина, а и минимальная.

При недостаче в поле данных вслед за ним в кадр вставляются дополнительные октеты-наполнители (значения стандартом не регламентированы).

Параметр MTU (Maximum Transmission Unit) определяет максимальный размер вкладываемых данных. Применительно к Ethernet, если значение поля Length/Type больше либо равно 1536 (600h), то указывает тип инкапсулируемых данных.

При необходимости, октеты-расширители дополняет кадр до тайм-слота (только в полу duplexном режиме).

6.0.3.10

Ethernet-заголовок имеет фиксированную длину.

Но, поскольку многие базирующиеся на Ethernet технологии (например, виланы) имеют собственные подзаголовки, заголовок, а следовательно и весь кадр, может увеличиться, правда незначительно и не затрагивая MTU (такие кадры иногда называют *baby giant*).

Некоторые технологии предусматривают значительное увеличение кадра уже за счет увеличения MTU. Например, параметр MTU технологии FCoE (Fibre Channel over Ethernet) равен 2500 байтам (такие кадры иногда называют *mini jumbo*).

Наконец, многие производители оборудования Ethernet предусмотрели нестандартное (но в большинстве случаев совместимое) административное увеличение MTU вплоть до 9000 байтов -- в первую очередь, для оптимизации пересылки больших объемов данных. Такие Ethernet-кадры называют *гигантскими (jumbo)*.

6.0.3.11

В качестве контрольного кода используется код CRC.

Предусмотрен контроль потока, работающий по принципу XON/XOFF.

6.0.3.12

При функционировании с пропускной способностью выше 100 Mbit/s (только в полудуплексном режиме) реализация может опционально передавать серию кадров ослабив контроль среды.

Такой режим работы называют *пакетным режимом* (burst mode).

Сразу после успешной передачи первого кадра начинается безусловная передача последующих кадров -- это возможно, поскольку передатчики других станций по-прежнему будут находиться в состоянии ожидания.

Интервалы между кадрами (interframe gaps), без которых принимающая станция вообще не сможет различать кадры, укорочены до минимума с помощью октетов-расширителей.

Количество кадров в пакете ограничено.

Считается, что в правильно сконфигурированном сегменте при передаче второго и последующих кадров пакета коллизии возникать не должны. Однако, если такая коллизия возникает, то она обрабатывается особо (выход из алгоритма с ошибкой) -- это так называемая поздняя коллизия (late collision).

6.0.3.13

Под автосогласованием физического уровня Ethernet понимают автоматическое определение максимальной скорости обмена данными и поддержки полнодуплексности (собственно auto-negotiation), а также варианта кабеля («прямой» либо кросс) (auto-MDI/MDIX) и некоторых других параметров (реализуют редко).

Автосогласование осуществляется последовательностями импульсов фиксированной длины, называемых FLPs (Fast Link Pulses), с минимальной скоростью (10 Mbit/s).

Из-за недостаточной стандартизации, во многих случаях оборудование разных производителей выполняет автосогласование с ошибками.

6.0.3.14

Parameters	MAC data rate			
	Up to and including 100 Mb/s	1 Gb/s	10 Gb/s	40 Gb/s and 100 Gb/s
slotTime	512 bit times	4096 bit times	not applicable	not applicable
interPacketGap ^a	96 bits	96 bits	96 bits	96 bits
attemptLimit	16	16	not applicable	not applicable
backoffLimit	10	10	not applicable	not applicable
jamSize	32 bits	32 bits	not applicable	not applicable
maxBasicFrameSize	1518 octets	1518 octets	1518 octets	1518 octets
maxEnvelopeFrameSize	2000 octets	2000 octets	2000 octets	2000 octets
minFrameSize	512 bits (64 octets)	512 bits (64 octets)	512 bits (64 octets)	512 bits (64 octets)
burstLimit	not applicable	65 536 bits	not applicable	not applicable
ipgStretchRatio	not applicable	not applicable	104 bits	not applicable

Примеры значений параметров Ethernet [IEEE]

6.0.4.1

Еще одним примером случайных методов доступа к моноканалу является гораздо более сложный алгоритм CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) -- множественный доступ с прослушиванием несущей и избеганием коллизий, описанный в стандарте Wi-Fi (IEEE 802.11).

6.0.4.2а

Для понимания алгоритма необходимо ввести термины из стандарта.

Применительно к Wi-Fi, MAC-подуровень канального уровня поделен еще на два слоя.

На нижнем слое расположен только один блок под названием DCF (Distributed Coordination Function) -- функционал распределенного координируемого взаимодействия. DCF и составляет ядро алгоритма CSMA/CA. Все станции сегмента должны поддерживать DCF.

Над DCF расположены:

1. PCF (Point Coordination Function) -- функционал координируемого взаимодействия с использованием станции-координатора.
2. HCF (Hybrid Coordination Function) -- функционал гибридного координируемого взаимодействия.
3. MCF (Mesh Coordination Function) -- функционал сеточного координируемого взаимодействия.

6.0.4.2b

Из них формируют следующие опциональные блоки:

1. PCF.
2. HCCA (HCF Controlled Access).
3. EDCA (HCF/MCF Contention Access).
4. MCCA (MCF Controlled Access).

Кроме DCF, наибольший интерес представляет PCF. Остальные блоки предназначены для поддержки QoS.

6.0.4.3

В настоящее время реализации Wi-Fi на физическом уровне (беспроводные) очень разнообразны -- используются до десяти различных способов модуляции.

Более того, для Wi-Fi характерно создание большого числа параллельных каналов.

6.0.4.4

Стандартом предусмотрены целых шесть вариантов отслеживаемых межкадровых интервалов -- IFSes (InterFrame Spaces):

1. RIFS (Reduced IFS) -- сокращенный.
2. SIFS (Short IFS) -- короткий.
3. PIFS (PCF IFS) -- для PCF.
4. DIFS (DCF IFS) -- для DCF.
5. AIFS (Arbitration IFS) -- для QoS-арбитража.
6. EIFS (Extended IFS) -- расширенный.

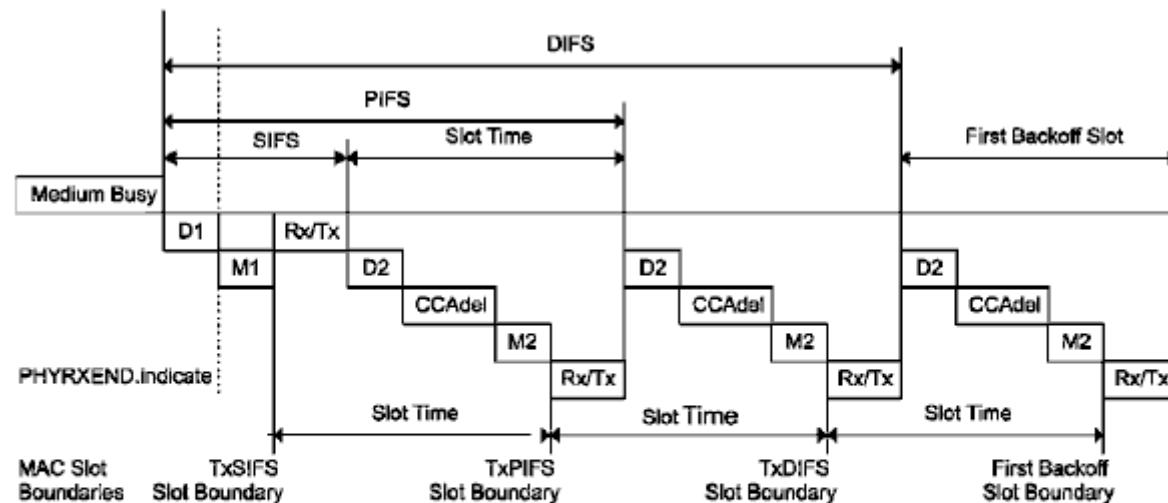
6.0.4.5

Для чего может понадобиться вводить разные IFSes?

6.0.4.6

Отслеживание различных IFSes в различных ситуациях влияет на способность станции «видеть щели» между кадрами, а значит и на способность «вклиниваться» в пересылку.

IFSes рассчитывают на основании комплекса параметров.



D1 = aRxRFDelay + aRxPLCPDelay (referenced from the end of the last symbol of a frame on the medium)

D2 = D1 + Air Propagation Time

Rx/Tx = aRXTXTurnaroundTime (begins with a PHYTXSTART.request)

M1 = M2 = aMACProcessingDelay

CCAdel = aCCA Time - D1

Кроме интервала DIFS, используемого функционалом DCF, наиболее интересны SIFS и PIFS.

[IEEE]

6.0.4.7

Случайную задержку измеряют в слот-таймах, как и в Ethernet, но алгоритм другой. Количество слот-таймов является случайным целым числом *Random*:

$$0 \leq Random \leq CW,$$

где *CW* (contention window) -- так называемое окно состязаний:

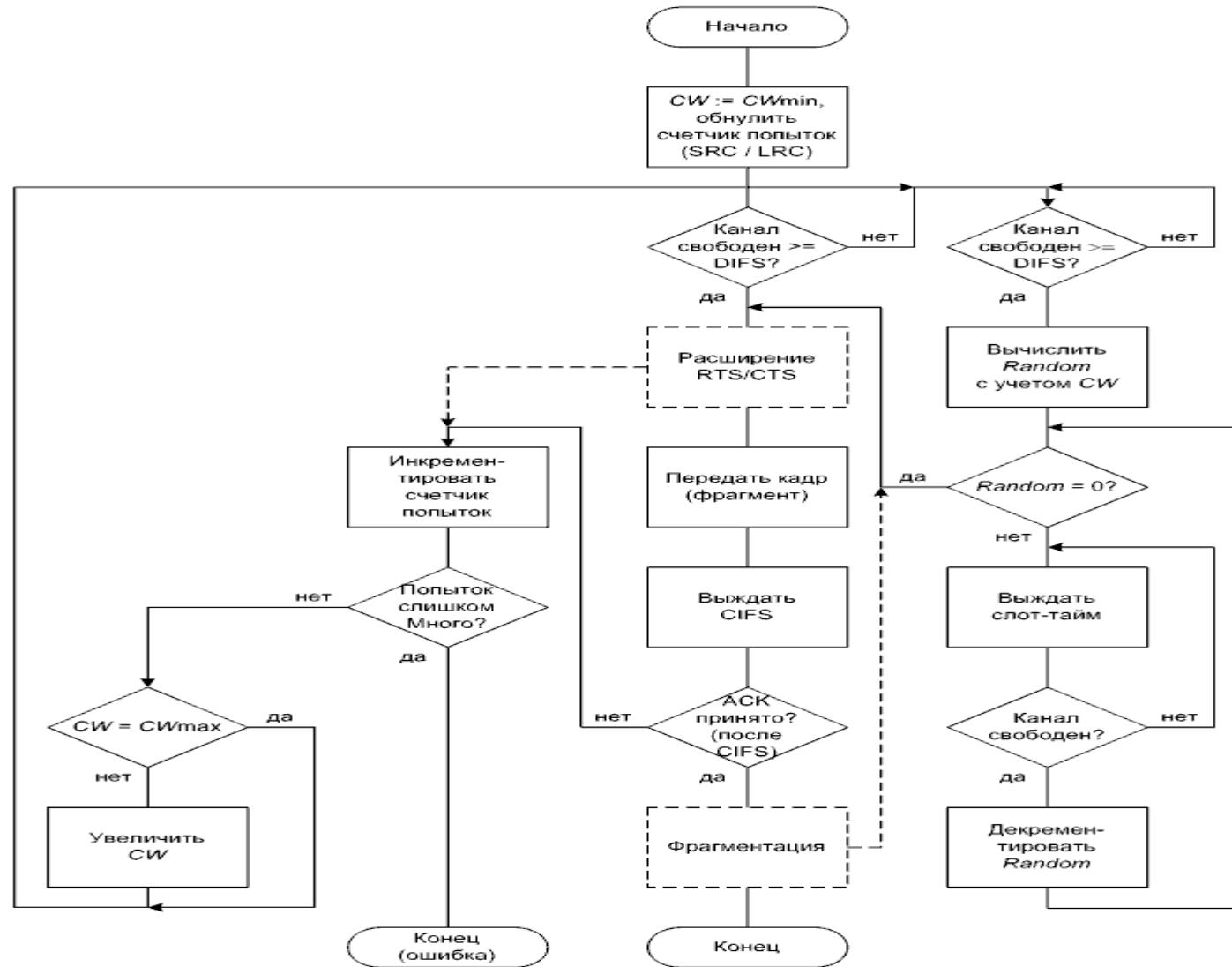
$$CW_{\min} \leq CW \leq CW_{\max},$$

и берется из ряда: 7, 15, 31 ... (два в некоторой степени минус один).

Крайние значения зависят от способа модуляции (типичное значение CW_{\min} -- 15, типичное значение CW_{\max} -- 1023).

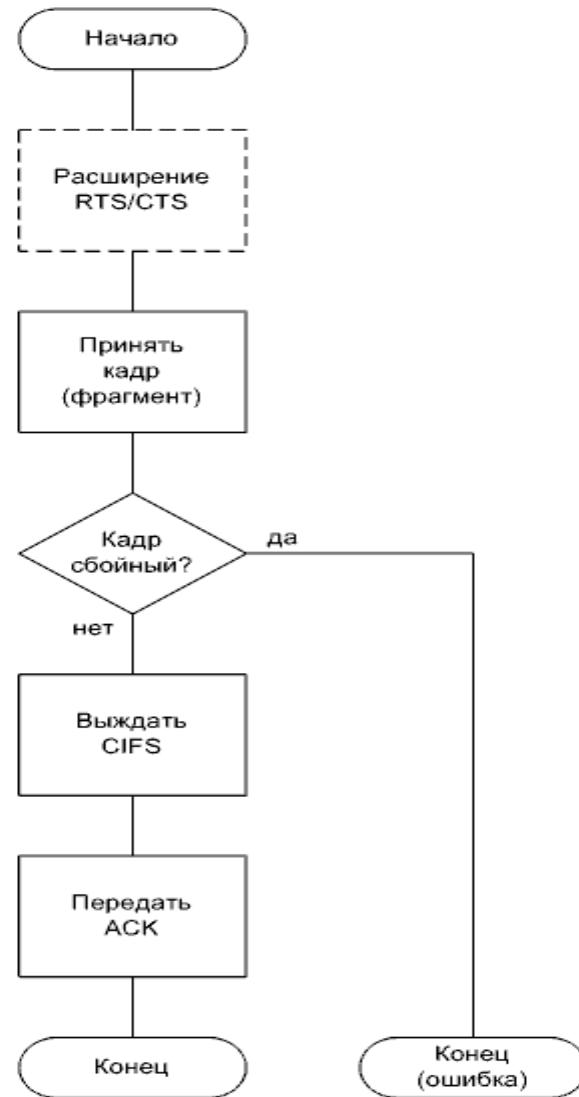
Предусмотрены два счетчика попыток: SRC (Short Retry Count) и LRC (Long Retry Count). Количество попыток ограничено. Выбор значения зависит от физического уровня.

6.0.4.8



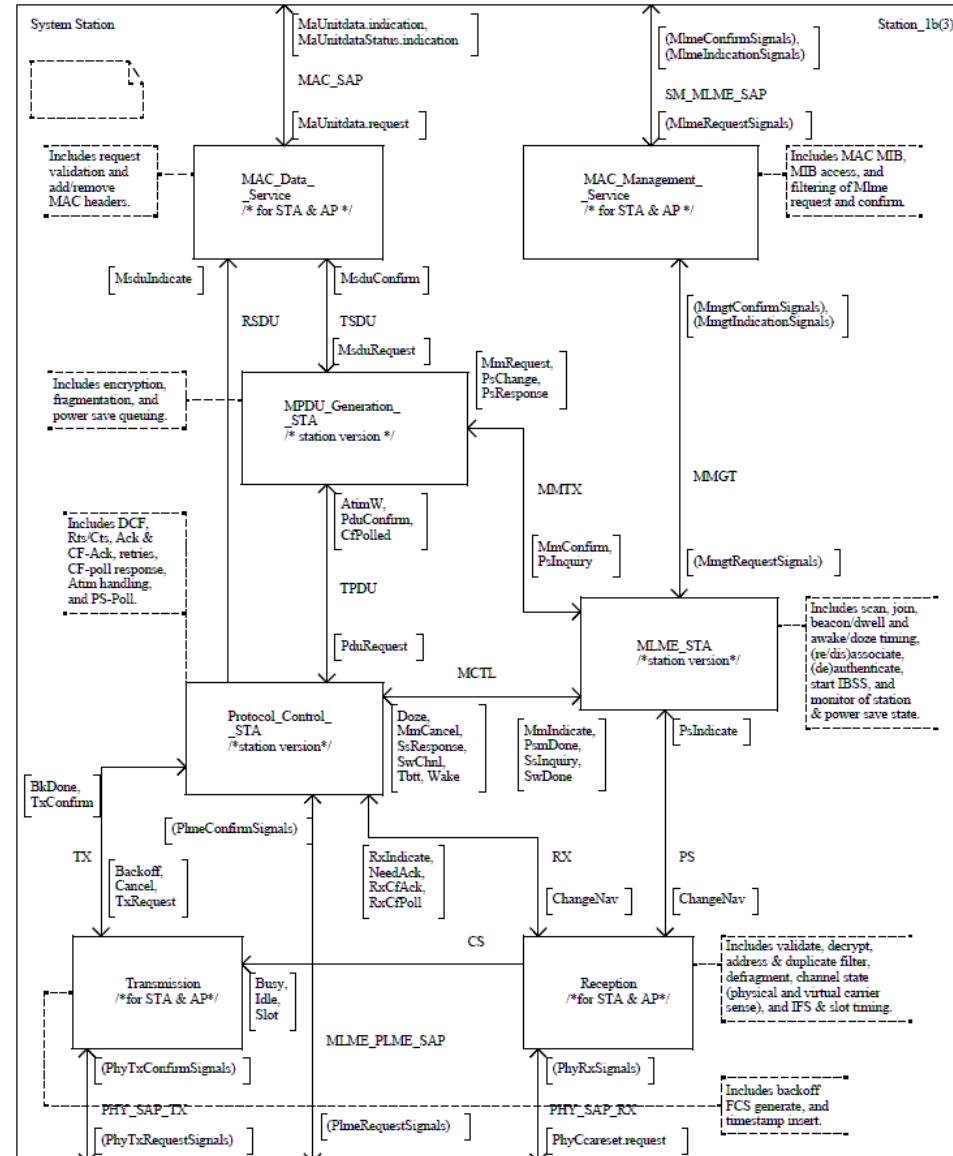
Очень упрощенный алгоритм CSMA/CA (Wi-Fi). Передача очередного кадра

6.0.4.9



Очень упрощенный алгоритм CSMA/CA (Wi-Fi). Прием очередного кадра

6.0.4.10



Фрагмент полного алгоритма Wi-Fi (одна страница из 192) [IEEE]

6.0.4.11

Для беспроводных каналов свойственны две проблемы, которые получили следующие названия:

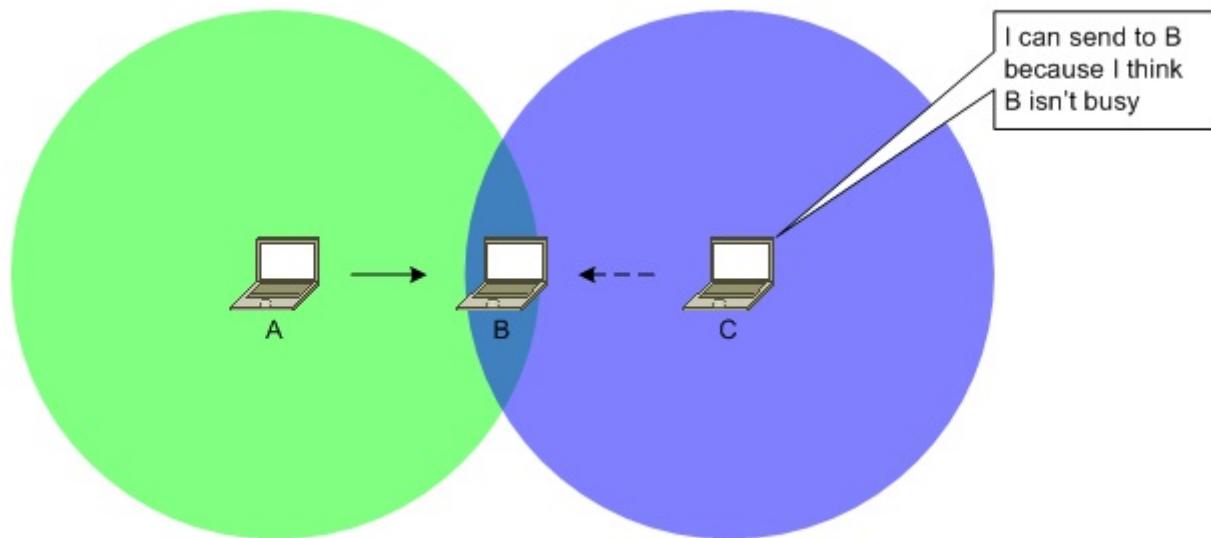
1. Hidden node problem -- проблема скрытой станции.
2. Exposed node problem -- проблема доступной станции.

Предполагается, что все станции взаимодействуют в рамках одного канала.

(Эти проблемы возникнут и в проводных каналах, если не учесть окно коллизий.)

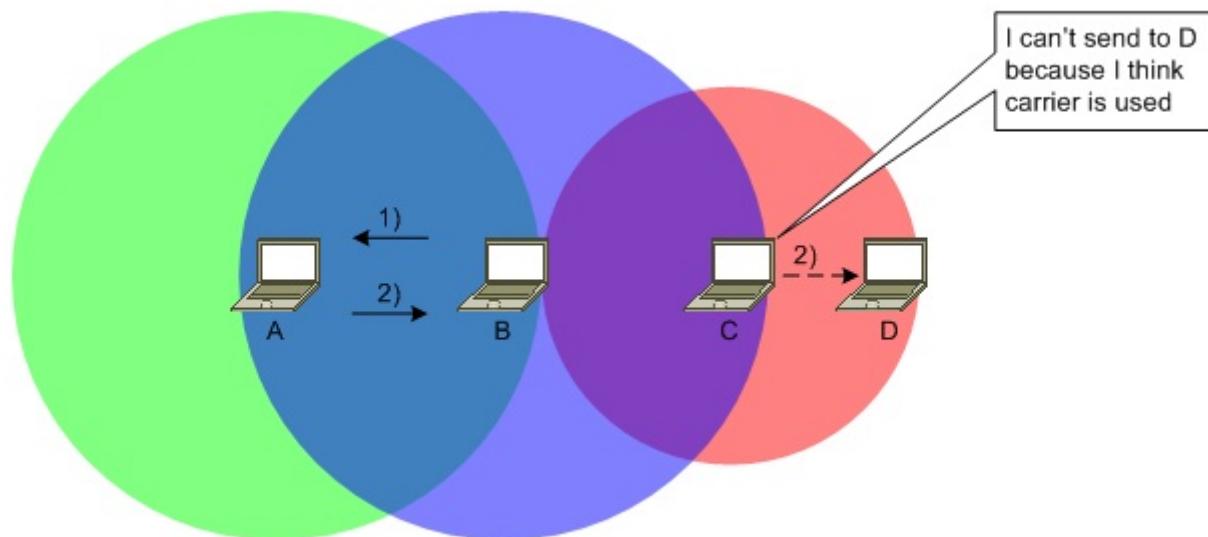
6.0.4.12

Проблему скрытой станции можно сформулировать так: станция С может ошибочно начать передачу станции В, так как не может «услышать» что станция А уже передает станции В (станция А «скрыта» от станции С).



6.0.4.13

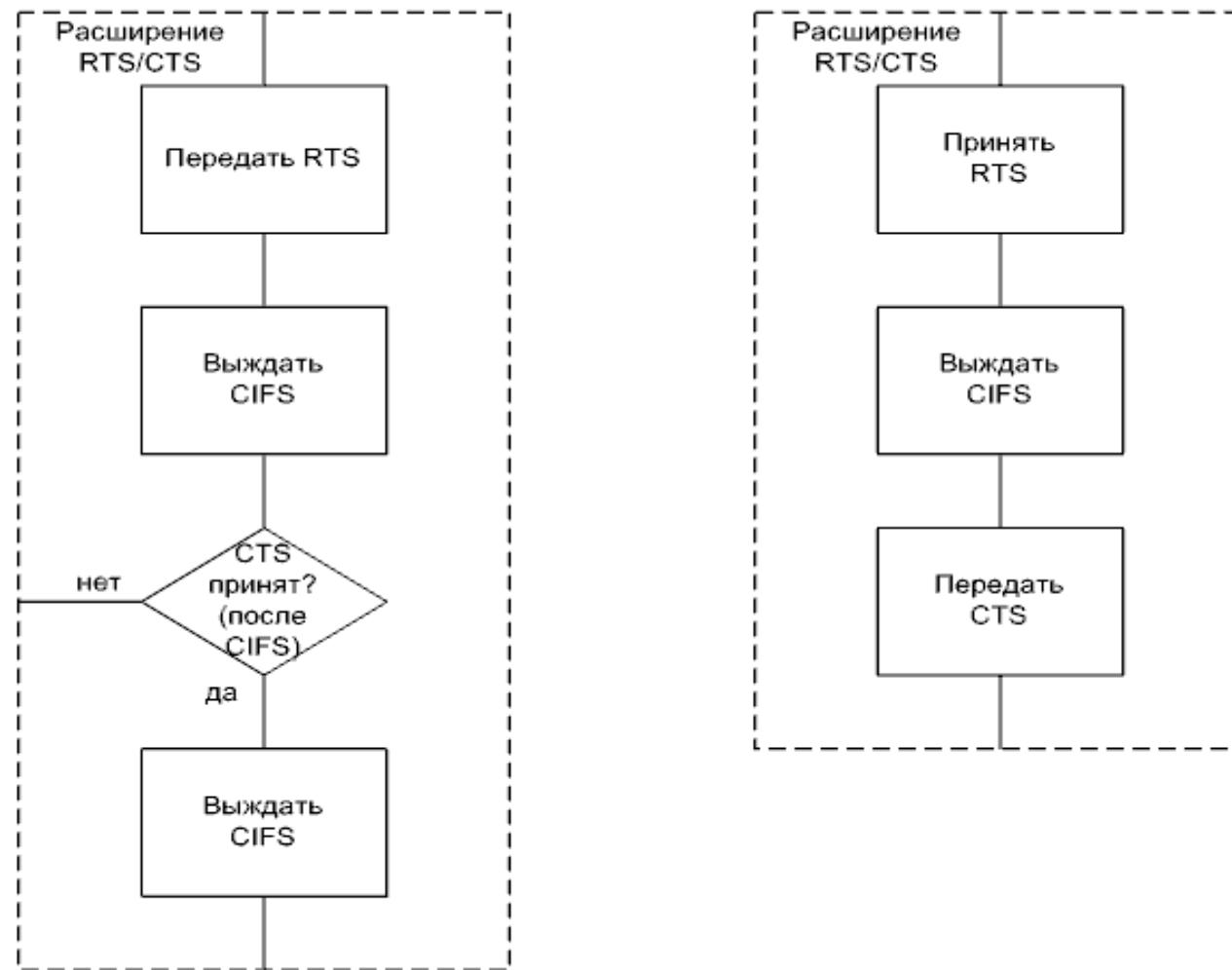
Проблему доступной станции можно сформулировать так: станция С, зная о взаимодействии станций А и В, не может передать станции D во время пассивности станции В, а могла бы, поскольку считает канал занятым ошибочно (станция С «доступна» для станции D).



6.0.4.14

Частично решить проблемы помогает опциональное расширение RTS/CTS.

6.0.4.15



Алгоритм CSMA/CA (Wi-Fi). Расширение RTS/CTS

6.0.4.16

Кадры, поступившие на канальный уровень для дальнейшей передачи, называют MSDUs (MAC Service Data Units). Они могут быть размером до 2304 байтов.

MSDUs разбиваются на меньшие фрагменты, называемые MPDUs (MAC Protocol Data Units), которые передаются в пакетном режиме. Длина фрагментов также ограничена (например, 4095 байтов).

Уменьшение длины фрагментов приводит к уменьшению вероятности коллизии при передаче отдельно взятого фрагмента, но и к увеличению количества фрагментов.

6.0.4.17



Алгоритм CSMA/CA (Wi-Fi). Фрагментация

6.0.4.18

Еще одним механизмом Wi-Fi для предотвращения коллизий является резервирование канала.

Все станции в сегменте обязаны иметь таймеры, называемые NAVs (Network Allocation Vectors). Каждый раз при резервировании значение таймера обновляется согласно временно'му интервалу резервирования и затем уменьшается.

Станция не имеет права начать передачу до тех пор, пока значение не достигнет нуля (плюс DIFS).

Обращение к таймеру при необходимости передать кадр, в терминологии Wi-Fi, называют виртуальным прослушиванием несущей (происходит параллельно с физическим прослушиванием).

6.0.4.19

Несмотря на все описанные меры, вероятность коллизий все-равно не равна нулю.

В связи с особенностями беспроводных каналов, в них передатчикам значительно сложнее самостоятельно обнаруживать коллизии. Поэтому эту функцию с них снимают и возлагают на приемники.

Вместо обнаружения коллизии, передатчик ждет положительное подтверждение ACK от приемника. Коллизия, как и любая другая проблема с кадром, приведет к отсутствию подтверждения и, далее, к повторной передаче.

6.0.4.20а

Формат кадра Wi-Fi так же сложен -- в сравнении с форматом кадра Ethernet. При этом наличие и названия последующих полей зависят от значения предыдущих.

2 Bytes	2 B	6 B	6 B	6 B	2 B	6 B	2 B	4 B	0 – 11454 B	4 B
Frame Control	Duration /ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control	HT Control	Frame Body	FCS
Header										
2 bits	2 b	4 b	1 b	1 b	1 b	1 b	1 b	1 b	1 b	1 b
Protocol Version	Type	Subtype	To DS	From DS	More Fragments	Retry	Power Management	More Data	Protected Frame	Order

Обобщенный формат кадра Wi-Fi

6.0.4.20b

Поля:

1. Frame Control -- контроль кадра.
2. Duration/ID -- длительность-идентификатор (0 -- 32767 us при резервировании канала, трактовка зависит например от наличия QoS).
3. Address 1 -- адрес 1.
4. Address 2 -- адрес 2.
5. Address 3 -- адрес 3.
6. Sequence Control -- контроль последовательности.
7. Address 4 -- адрес 4.
8. QoS Control -- контроль QoS.
9. HT Control (High Throughput) -- контроль интенсивной пересылки (при QoS).
10. Frame Body -- содержимое кадра (данные).
11. FCS (Frame Control Sequence) -- контрольная сумма.

Обобщенный формат кадра Wi-Fi

6.0.4.20c

Поля контроля кадра:

1. Protocol Version -- версия протокола (до сих пор равна нулю).
2. Type -- тип: 00b -- Management -- управление, 01b -- Control – контроль (фактически, более низкоуровневое управление), 10b -- Data -- данные, 11b -- Reserved -- зарезервировано.
3. Subtype -- подтип (в настоящее время определено около сорока подтипов).
4. To DS -- флаг направления в распределительную систему (проводную систему, связывающую беспроводные сегменты).
5. From DS -- флаг направления из распределительной системы.
6. More Fragments -- флаг наличия фрагментации.
7. Retry -- флаг повторной попытки передачи.
8. Power Management -- флаг режима энергосбережения.
9. More Data -- флаг наличия дополнительных данных (например, буферизированных данных для находящейся в режиме энергосбережения станции).
10. Protected Frame -- флаг защищенности кадра (шифрования).
11. Order -- флаг упорядоченности (при QoS).

6.0.4.20d

Таким образом, существуют три типа кадров.

В зависимости от подтипа кадра в адресных полях могут комбинироваться до четырех из пяти возможных адресов:

BSSID (Basic Service Set Identifier) -- идентификатор так называемой базовой зоны обслуживания (то есть беспроводного сегмента),

SA (Source Address) -- адрес источника,

DA (Destination Address) -- адрес назначения,

TA (Transmitting station Address) -- адрес станции-передатчика (непосредственного),

RA (Receiving station Address) -- адрес станции-приемника (непосредственного).

6.0.4.21

Table Valid type and subtype combinations

Type value b3 b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
00	Management	0000	Association request
00	Management	0001	Association response
00	Management	0010	Reassociation request
00	Management	0011	Reassociation response
00	Management	0100	Probe request
00	Management	0101	Probe response
00	Management	0110	Timing Advertisement
00	Management	0111	Reserved
00	Management	1000	Beacon
00	Management	1001	ATIM
00	Management	1010	Disassociation
00	Management	1011	Authentication
00	Management	1100	Deauthentication
00	Management	1101	Action
00	Management	1110	Action No Ack
00	Management	1111	Reserved
01	Control	0000–0110	Reserved
01	Control	0111	Control Wrapper
01	Control	1000	Block Ack Request (BlockAckReq)
01	Control	1001	Block Ack (BlockAck)
01	Control	1010	PS-Poll
01	Control	1011	RTS
01	Control	1100	CTS
01	Control	1101	ACK
01	Control	1110	CF-End
01	Control	1111	CF-End + CF-Ack
10	Data	0000	Data
10	Data	0001	Data + CF-Ack
10	Data	0010	Data + CF-Poll
10	Data	0011	Data + CF-Ack + CF-Poll
10	Data	0100	Null (no data)
10	Data	0101	CF-Ack (no data)
10	Data	0110	CF-Poll (no data)
10	Data	0111	CF-Ack + CF-Poll (no data)
10	Data	1000	QoS Data
10	Data	1001	QoS Data + CF-Ack
10	Data	1010	QoS Data + CF-Poll
10	Data	1011	QoS Data + CF-Ack + CF-Poll
10	Data	1100	QoS Null (no data)
10	Data	1101	Reserved
10	Data	1110	QoS CF-Poll (no data)
10	Data	1111	QoS CF-Ack + CF-Poll (no data)
11	Reserved	0000–1111	Reserved

Типы и подтипы кадров Wi-Fi [IEEE]

6.0.4.22

При PCF, «привязка» оконечной станции к станции-координатору (так называемой точке доступа) протекает в три фазы:

1. Discovery -- обнаружение.
2. Authentication -- аутентификация.
3. Association -- ассоциирование.

6.0.4.23

Обнаружение может быть активным и пассивным.

Оконечная станция в поисках станции-координатора может сканировать эфир пассивно, а может активно генерировать кадры-«пробы» (probe requests).

Со своей стороны, станция-координатор может отвечать на кадры-«пробы» (probe responses) пассивно, а может активно периодически извещать о себе и своих услугах с помощью кадров-«маяков» (beacons).

6.0.4.24

В рамках CSMA/CA существуют две группы алгоритмов:

1. Без наличия станции-координатора и с упреждающим ют-сигналом.
2. С наличием станции-координатора.

Упреждающий ют-сигнал не только информирует о намерении передать кадр, а и является признаком коллизии. Если в процессе передачи своего кадра станция распознает ют-сигнал от другой станции, то возникла коллизия.

6.0.5.1

Кроме Ethernet и Wi-Fi, в список существующих реализаций случайных методов следует включить технологию Aloha с одноименным алгоритмом.

Эта технология была разработана в университете Гавайских островов и была одной из самых ранних технологий КС.

В виде стандарта так и не была утверждена.

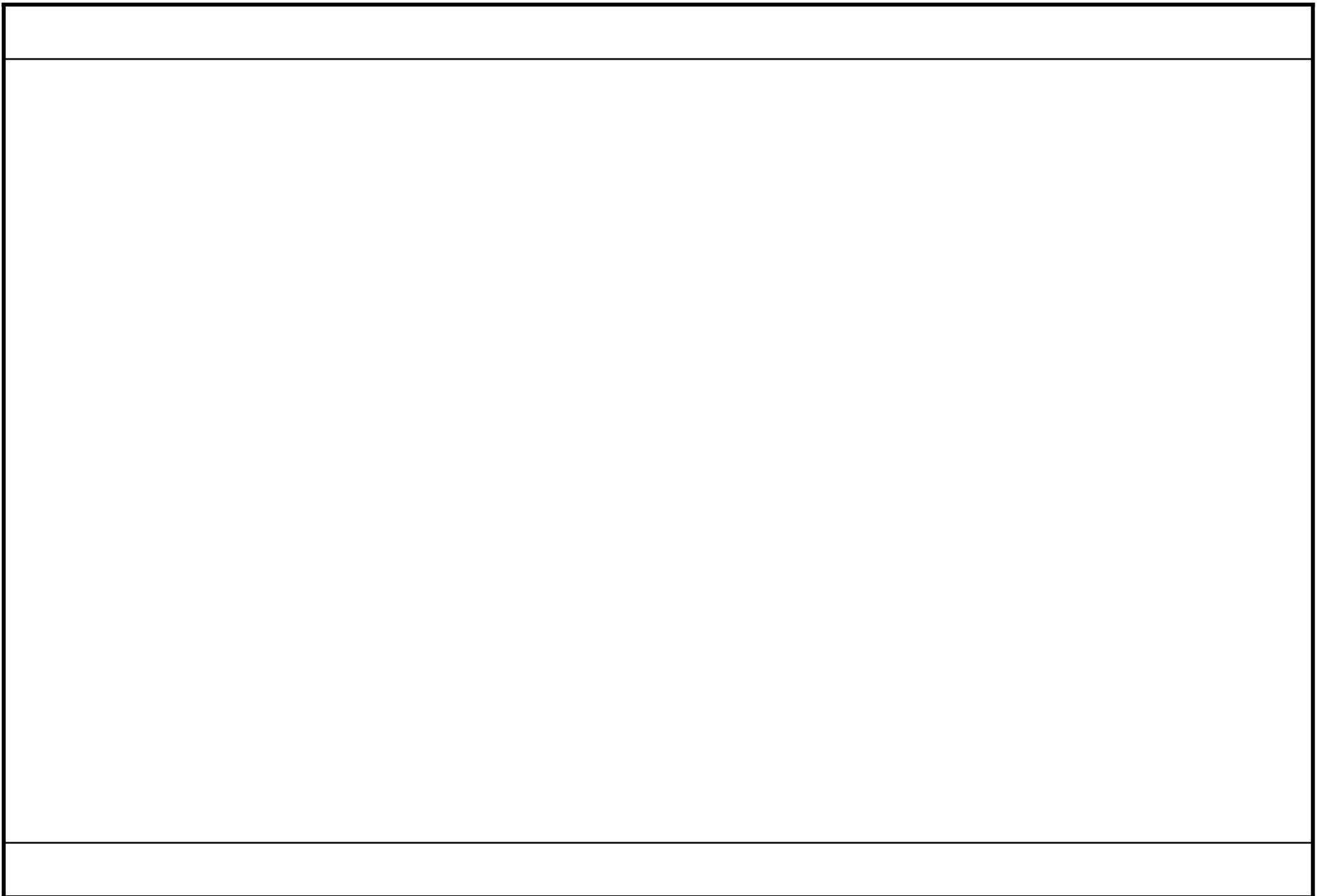
Нашла лишь ограниченное применение в беспроводных каналах для подключения мобильных телефонов первого поколения. Уже давно устарела.

Скорость: меньше 1 Mbit/s.

Логическая топология: двунаправленное кольцо с разделенными цепями передатчиков и приемников.

Физическая топология: звезда. Требовалось дополнительное сетевое оборудование (концентраторы).

Алгоритм представлял собой сильно упрощенный вариант алгоритма Ethernet. Позже был немного усовершенствован и получил название Slotted Aloha.



ДЕТЕРМИНИРОВАННЫЕ МЕТОДЫ ДОСТУПА К МОНОКАНАЛУ

7.0.1.1

При разработке любого из методов доступа к моноканалу один из базовых посылов -- это учет топологических особенностей сегмента.

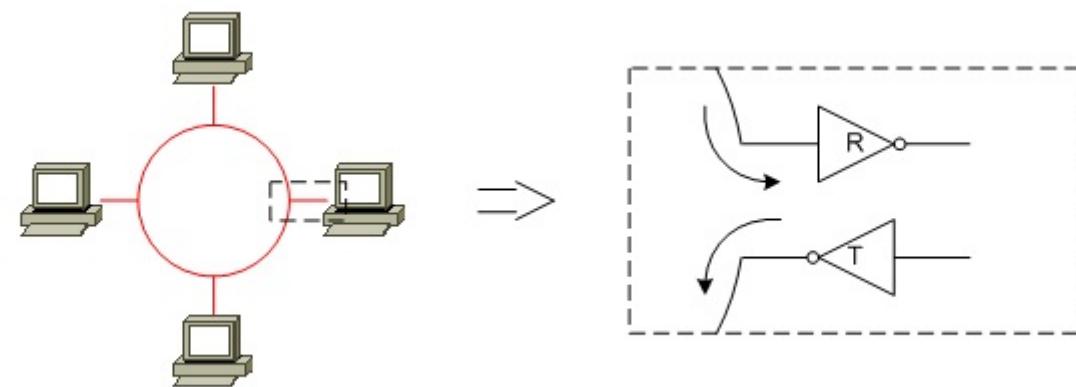
Если случайные методы уместно использовать при шинной топологии, применительно к которой четко выражена возможность возникновения коллизий, то детерминированные методы хорошо «ложатся» на кольцевую топологию.

Концептуальная разница между случайными и детерминированными методами заключается в том, возникает ли случайность при «обращении» станции к моноканалу.

7.0.1.2

Кольцо можно рассматривать как своеобразный моноканал, один такт работы которого представляет собой полный либо частичный «обход» кадром всех станций.

Более подробно типичную кольцевую топологию можно представить следующим образом.



7.0.1.3

Применительно к приведенной топологии, при доступе к моноканалу никаких проблем казалось бы возникать не должно.

Действительно, физические коллизии для такой схемы невозможны, но проявляется то, что можно назвать особым видом логических коллизий.

Если при некотором такте кольца какая-либо из станций имеет собственный кадр для передачи и при этом получила из кольца еще один кадр, который необходимо «продвигать» дальше, то появляется вопрос о том, какой из этих кадров передавать.

7.0.1.4

Частично противоречие может быть разрешено буферизацией кадров. Но возлагать на обычную пользовательскую станцию функции полноценного сетевого моста канального уровня крайне нецелесообразно. Кроме того, буферизация позволяет только «удерживать», то есть не терять кадры. Сугубо алгоритмический вопрос о том, какой же из кадров (имеющийся кадр для трансляции либо принятый кадр для ретрансляции) передавать раньше все-таки оставляет без ответа.

Единственным способом преодоления логических коллизий является *введение приоритетов (priorities)*.

В то время как все случайные методы «заязаны» на генератор случайных задержек, все детерминированные методы «заязаны» на систему приоритетов в том или ином виде.

Возникает задача распределенного либо централизованного назначения приоритетов, причем ни одна из станций кольца заранее ничего «не знает» о других станциях.

7.0.1.5

При использовании механизма приоритетов не обойтись без так или иначе выраженного арбитра.

В качестве арбитра может выступать специальный служебный кадр, который в русскоязычной литературе обычно называют *маркером* (*token*).

7.0.1.6

Таким образом, основные критерии классификации детерминированных методов:

- централизованное либо распределенное управление;
- алгоритм назначения приоритетов;
- топологические особенности.

7.0.1.7

На эффективность детерминированных методов наиболее существенное влияние оказывают те же факторы, что и в ситуациях со случайными методами:

- количество взаимодействующих станций;
- частота синхронизации;
- длина кадра.

7.0.1.8

Назовите основные достоинства детерминированных методов доступа.

Назовите основные недостатки детерминированных методов доступа.

7.0.1.9

Если сравнивать детерминированные методы со случайными, то сложно сказать какие из них «лучше». При применении случайных методов основные потери производительности возникают из-за вносимых задержек, а при применении детерминированных методов потери обусловлены ретрансляцией кадров.

Если оценивать реализации, которые уже имеются на рынке, то все же детерминированные алгоритмы в среднем демонстрируют бо'льшую производительность. Однако оборудование в среднем более дорогостоящее.

7.0.2.1

С точки зрения изучения детерминированных методов доступа к моноканалу наиболее наглядным примером является классический алгоритм, описанный в стандарте Token Ring (IEEE 802.5).

7.0.2.2

В Token Ring применяется централизованное управление.

Закономерным следствием является необходимость включения в кольцо по крайней мере одной управляющей станции, наделенной особыми полномочиями и призванной инициализировать кольцо и следить за его работоспособностью. В терминологии Token Ring такую управляющую станцию обобщенно называют *станцией-монитором* (monitor station).

Кроме единственной основной станции-монитора (active monitor) в состав кольца может входить некоторое количество резервных (standby monitors).

Функции станции-монитора:

1. Инициализировать подключившиеся к кольцу станции.
2. Тактировать (на физическом уровне) работу кольца.
3. Контролировать наличие и валидность маркера.
4. Предотвращать зацикливания.

7.0.2.3

В отличие от сегмента Ethernet, где все станции равноправны и действуют по одному и тому же алгоритму, в сегменте Token Ring предусмотрены станции нескольких видов.

Наряду с выделяемыми на канальном уровне станциями-мониторами, на более высоких уровнях рекомендуется выделять следующие станции:

1. System managers -- системные менеджеры (на них сосредоточены управляющие системой на основе Token Ring процессы).
2. Servers -- различные серверы (configuration report servers, ring error monitors, ring parameter servers).
3. Data stations -- информационные станции (обычные пользовательские станции).

Функциональное наполнение перечисленных видов станций выходит за рамки стандарта.

7.0.2.4

Не смотря на то, что теоретически кольцо предполагает некоторую возможность «распараллеливания» (то есть, одновременно по разным частям кольца могут циркулировать несколько кадров), очень обобщенно алгоритм Token Ring можно представить как «бесконечно» циркулирующий под управлением станции-монитора маркер, который анализируется всеми пользовательскими станциями и к которому при необходимости «цепляются» данные.

7.0.2.5

Для того чтобы понять заложенный в стандарт алгоритм, сначала необходимо рассмотреть форматы кадров Token Ring и назначение основных полей.

В стандарте предусмотрены четыре вида передаваемых последовательностей:

1. Token -- маркер.
2. Frame -- кадр.
3. Abort Sequence -- прерывающая последовательность.
4. Fill -- заполняющая последовательность.

Каждая из станций в любое время должна распознавать (и различать) маркеры, кадры и специальные последовательности.

7.0.2.6

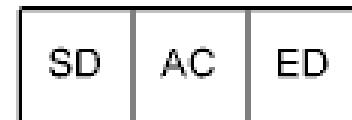
SFS			FCS Coverage					EFS		
SD	AC	FC	DA	SA	RI	INFO	FCS	ED	FS	IFG
Error (E-Bit) Coverage										

Поля:

- SD (Starting Delimiter) -- начальный разделитель.
- AC (Access Control) -- контроль доступа.
- FC (Frame Control) -- контроль кадра.
- DA (Destination Address) -- адрес назначения.
- SA (Source Address) -- адрес источника.
- RI (Routing Information) -- информация о маршрутизации (может отсутствовать).
- INFO (Information) -- данные (могут отсутствовать).
- FCS (Frame Check Sequence) -- контрольная сумма.
- ED (Ending Delimiter) -- конечный разделитель.
- FS (Frame Status) -- состояние кадра.
- IFG (InterFrame Gap) – межкадровый интервал.

Формат кадра Token Ring

7.0.2.7



Формат маркера Token Ring

7.0.2.8



Формат прерывающей последовательности Token Ring

7.0.2.9

SD и ED фактически являются флагами начала и конца кадра.
Между IFG и SD передается заполняющая последовательность.

7.0.2.10

С точки зрения алгоритма контроля доступа наибольший интерес представляет одноименное поле, а также поле состояния кадра.



Где:

P (Priority bits) -- текущий уровень приоритета.

T (Token bit) -- идентификатор маркера: 0 -- маркер, 1 -- кадр.

M (Monitor bit) -- бит монитора.

R (Reservation bits) -- запрашиваемый уровень приоритета.

Формат поля контроля доступа

7.0.2.11



Где:

- A (Address-recognized bit) -- флаг распознания адреса (дублируется).
- C (frame-Copied bit) -- флаг копирования кадра (дублируется).
- r (reserved) -- зарезервировано.

Формат поля состояния кадра

7.0.2.12

В стандарт заложена комплексная система приоритетов, однако некоторые «тонкости» оставлены на откуп реализациям.

Механизм приоритетов Token Ring **основан** на связке двух полей -- P и R.

Поле P отображает текущий уровень приоритета, а поле R -- запрашиваемый.

Каждое из этих полей может иметь значение от 000b до 111b, то есть доступно восемь уровней приоритета.

7.0.2.13

Условно можно выделить два режима взаимодействия:

1. Все станции имеют одинаковые приоритеты («отсутствие» приоритетов).
2. Станции могут иметь разные приоритеты («наличие» приоритетов, совместимое расширение первого режима, некоторые станции могут пользоваться кольцом более интенсивно, связь с QoS).

7.0.2.14

При «отсутствии» приоритетов станция-монитор создает и «запускает» в кольцо маркер с нулевыми полями Р и R (назначение этих полей не проявляется).

7.0.2.15а

С помощью маркера, который передается по цепочке от станции к станции, предоставляется право на передачу.

Если у станции нет своего кадра для передачи, то она передает маркер дальше.

Если у станции есть кадр для передачи, то она захватывает маркер, заменой значения поля T с нуля на единицу преобразует маркер в кадр, добавляет все соответствующие поля и передает.

Приоритет автоматически «достается» станции, до которой маркер дошел раньше.

Внесенный таким образом в кольцо кадр ретранслируется всеми промежуточными станциями до тех пор, пока не достигнет адресованной станции-абонента.

7.0.2.15b

За удаление кадра из кольца ответственна станция, создавшая его.

Поэтому станция-абонент, распознавшая свой адрес в принятом кадре, вместо удаления кадра отмечает факт распознавания присваиванием единичных значений обоим битам А и передает кадр дальше.

Если станция-абонент «забирает» данные из кадра, то она присваивает единичные значения и обоим битам С.

Значения битов А и С проверяются при возвращении кадра к создавшей его станции.

На основании результатов проверки делаются соответствующие выводы. Но нужно освободить маркер.

В нормальном случае станция освобождает маркер сразу после того, как дождется возвращения кадра.

7.0.2.16

Существует опциональная возможность освободить маркер более быстро.

При раннем освобождении маркера (early token release) сразу вслед за кадром передается новый маркер, а старый маркер не воссоздается.

В результате, несколько кадров смогут находиться в кольце одновременно (максимальное количество кадров будет равно максимальному количеству станций), но маркер всегда будет только один.

За счет того, что разные такты кольца «накладываются» друг на друга, потенциально можно получить значительный временной выигрыш.

Станции, не использующие и использующие раннее освобождение маркера, могут сосуществовать.

7.0.2.17

Владение маркером ограничено во времени и контролируется с помощью таймера ТНТ (Token Holding Time).

7.0.2.18

При наличии приоритетов взаимодействие значительно усложняется. В такой ситуации необходимо различать следующие приоритеты (применительно к каждой станции):

1. Pf (в стандарте Pm) -- уровень приоритета ожидающего передачи кадра (в общем случае станция может генерировать кадры с разными уровнями приоритета).
2. Pr -- значение поля R в принятом маркере либо кадре.
3. Rr -- значение поля R в принятом маркере либо кадре.
4. Pt (более «узко» в стандарте определен как Px) -- значение поля R, которое будет записано в передаваемый маркер либо кадр ($Px = \max(Pm, Rr)$).
5. Rt (в стандарте явно не определен) -- значение поля R, которое будет записано в передаваемый маркер либо кадр.
- +6. Sr -- текущее значение сохраненного Pr (вершина стека LIFO).
- +7. St (в стандарте Sx) -- текущее значение сохраненного Pt (вершина стека LIFO).

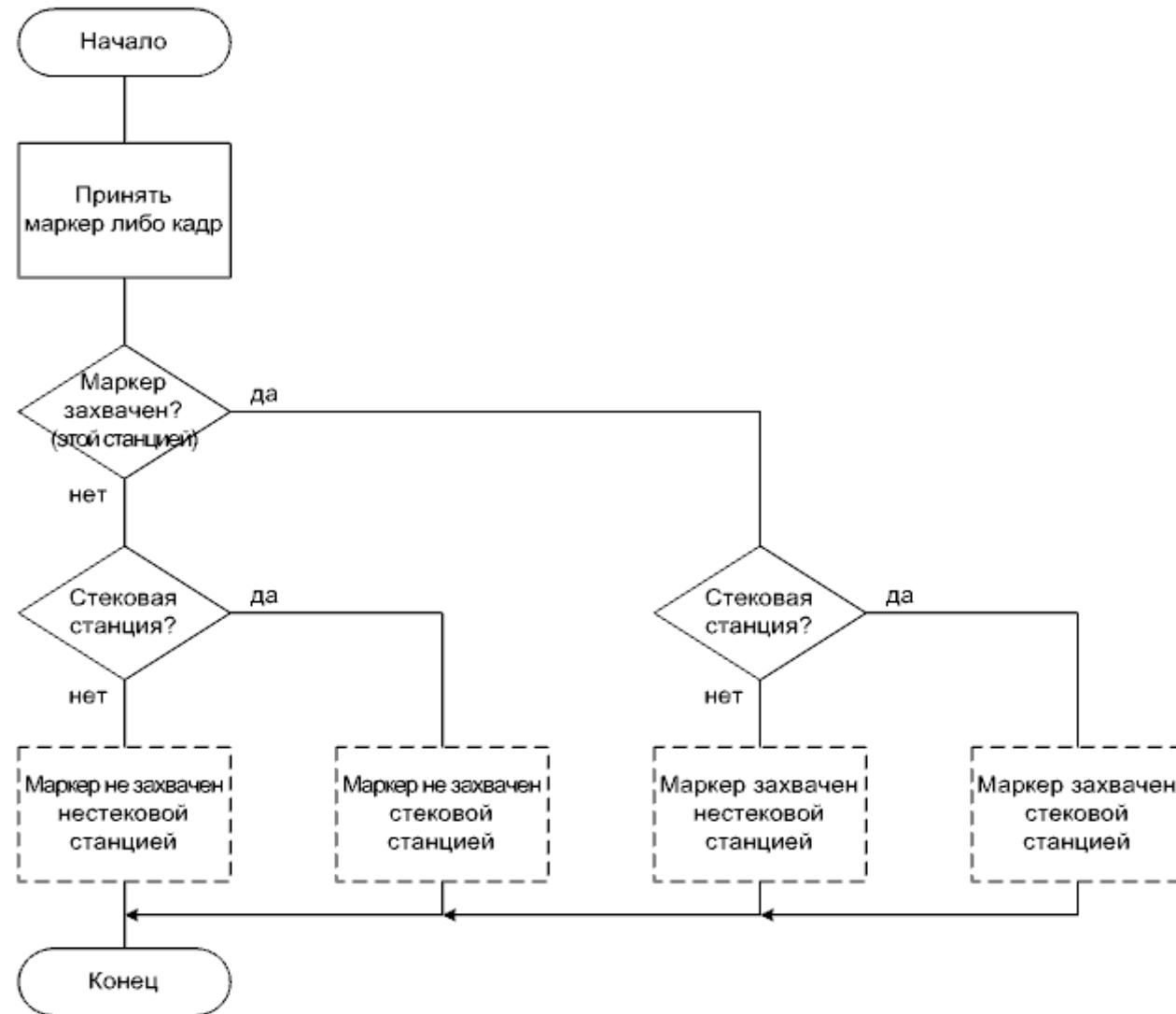
7.0.2.19

Рассмотрим IBM-совместимую модель приоритетов (стандарт создавался на базе разработки IBM, как и стандарт Ethernet создавался на базе разработки Xerox).

С точки зрения отдельно взятой станции порядок доступа к кольцу можно свести к трем шагам:

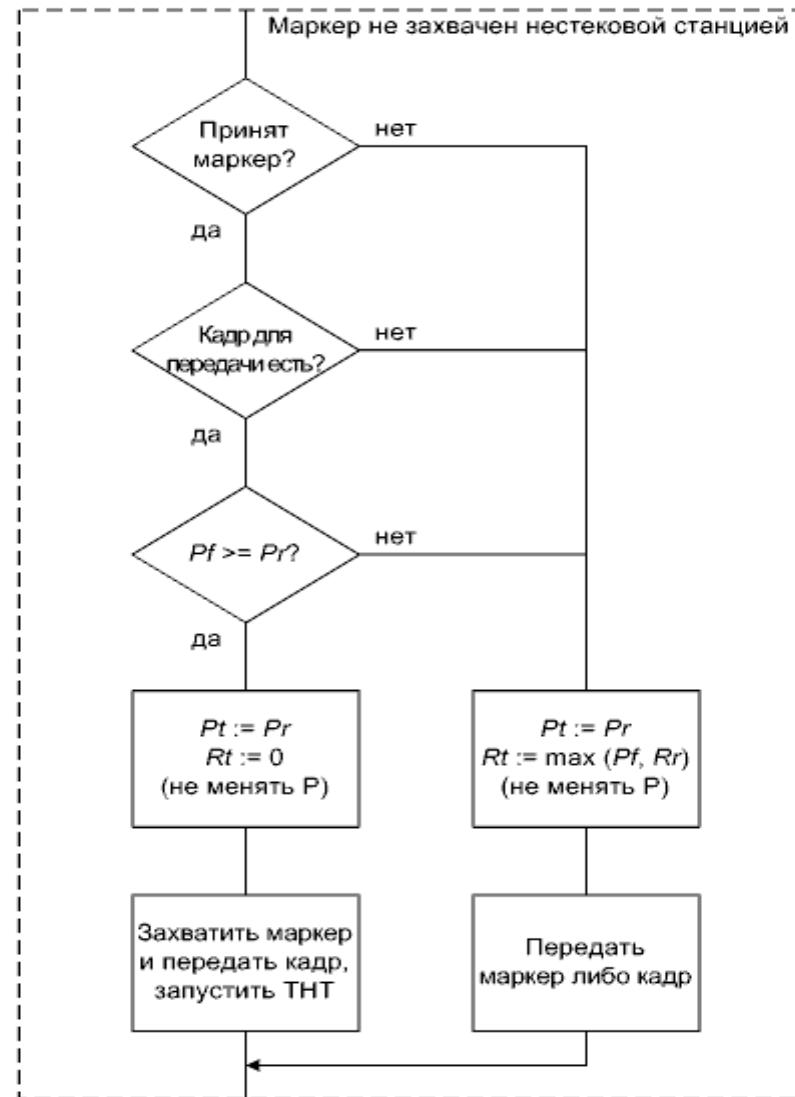
1. Захват маркера и передача кадра.
2. Освобождение маркера и при необходимости коррекция текущего уровня приоритета.
3. Восстановление текущего уровня приоритета если он был скорректирован.

7.0.2.20



Алгоритм Token Ring. Приоритеты

7.0.2.21



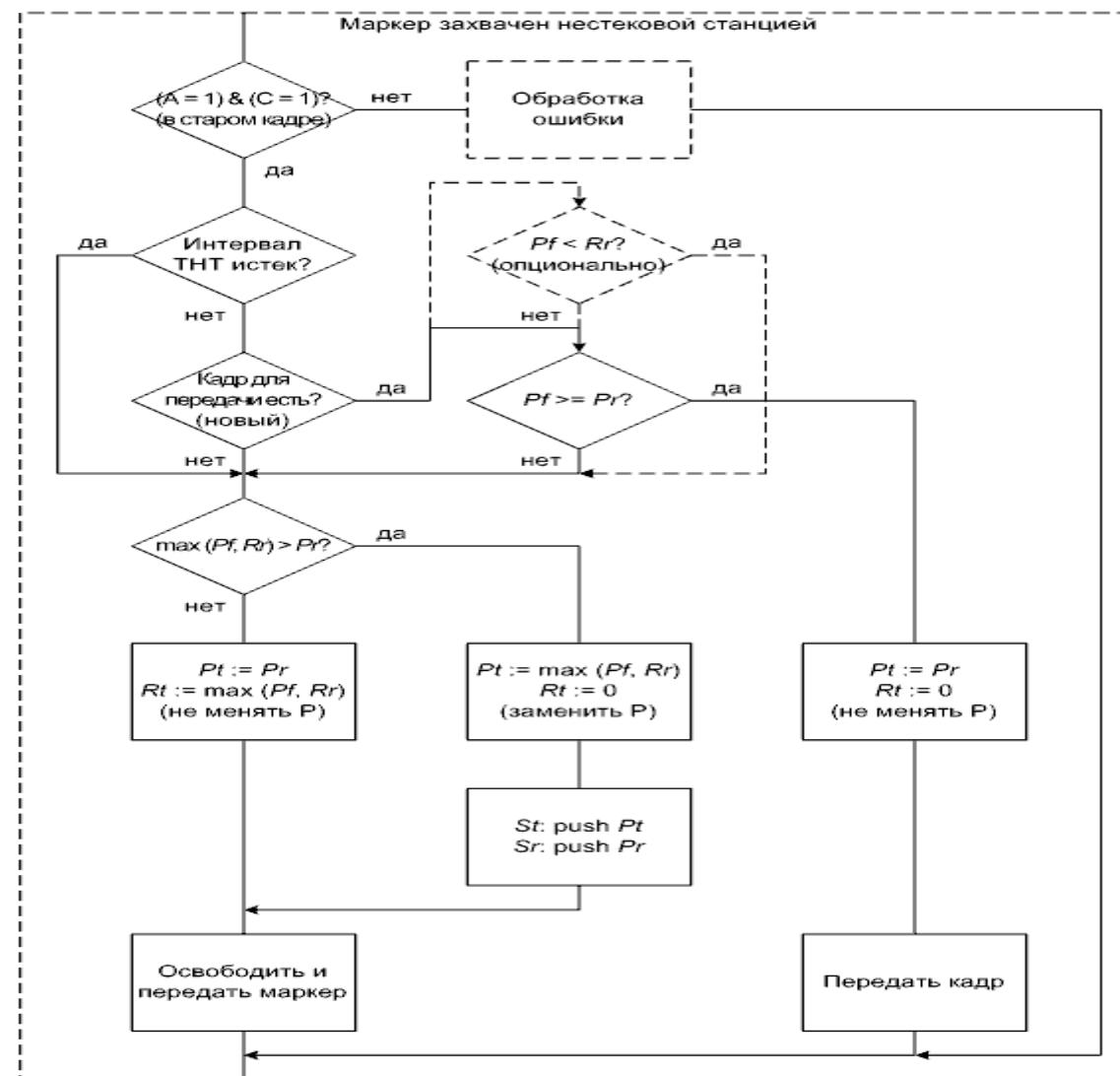
Алгоритм Token Ring. Приоритеты. Маркер не захвачен нестековой станцией

7.0.2.22

Кроме уровня приоритета станции, точнее кадра на станции, имеет место уровень приоритета в кольце, то есть текущий уровень приоритета.

Такт кольца начинается с захвата маркера некоторой станцией. При захвате маркера уровень приоритета в кольце не меняется.

7.0.2.23



Алгоритм Token Ring. Приоритеты. Маркер захвачен нестековой станцией

7.0.2.24а

Для того чтобы в кольце с конкуренцией наиболее приоритетный кадр либо кадры, которые находятся на одной либо нескольких станциях, передать в первую очередь необходимо повысить приоритет в кольце до соответствующего уровня. При этом нужно учесть все возникшие запросы.

Для того чтобы отследить запросы нужно «обойти» кольцо полностью. Для дальнейшего продвижения всегда выбирается запрос о наибольшем уровне. Так что запросы могут теряться.

Вполне логично совместить обход с первым либо очередным тактом кольца и новый уровень приоритета в кольце установить именно при освобождении маркера станцией. Данная схема не требует вмешательства станции-монитора. Если бы уровень приоритета в кольце меняла станция-монитор, то реакция была бы более замедленной. Станция-монитор, как и при «отсутствии» приоритетов, только создает маркер с нулевыми полями Р и R и контролирует его.

7.0.2.24b

Понятно, что изначально уровень приоритета в кольце должен быть минимальным.

Станция повышает приоритет в кольце для себя или для других станций (одной либо нескольких).

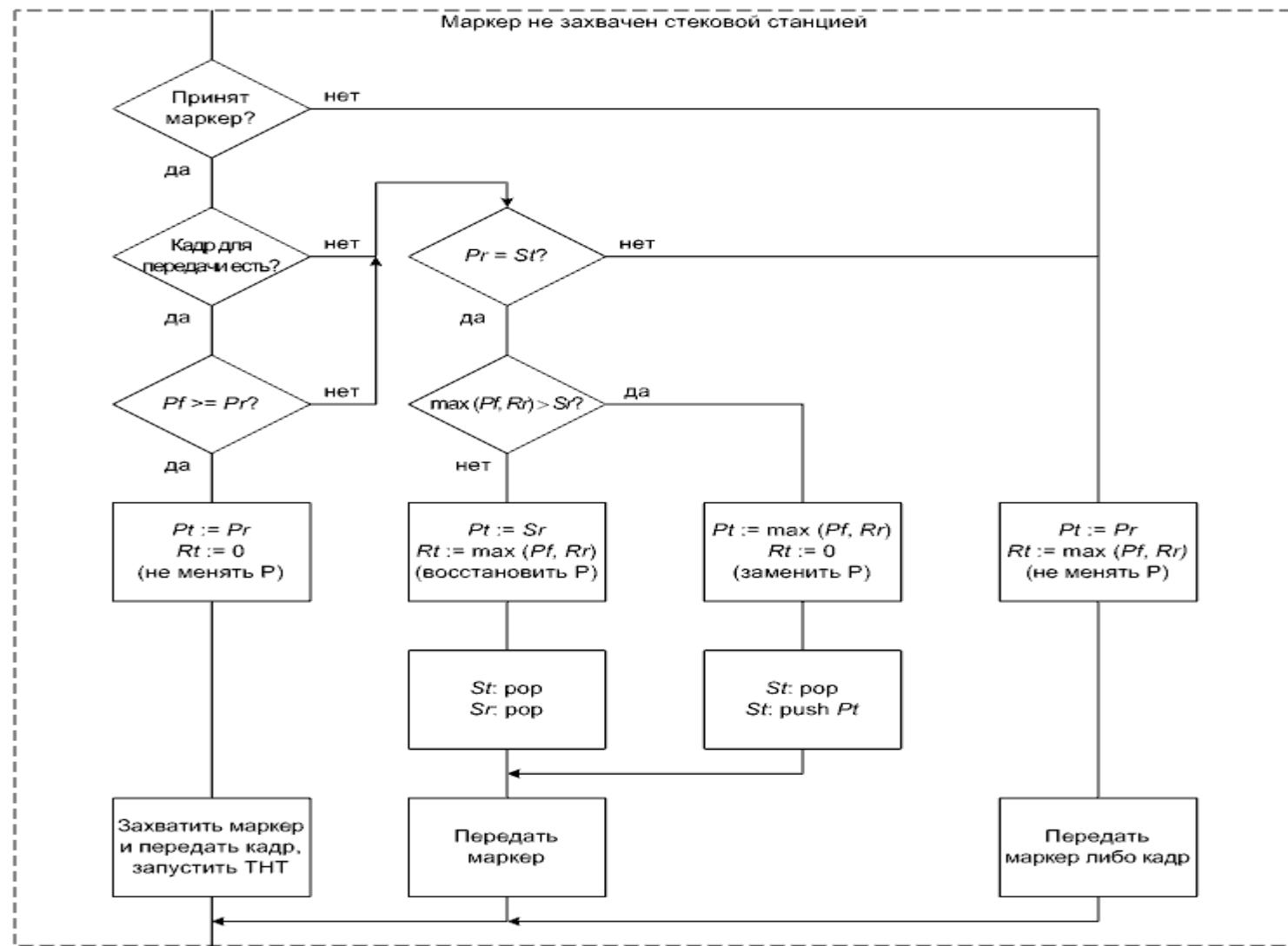
Приоритет в кольце может быть поднят сразу через несколько уровней.

Одна станция может повысить приоритет в кольце несколько раз.

При всем этом получается, что до одного уровня приоритет в кольце может быть повышен только одной станцией.

При замене уровня приоритета в кольце необходимо сохранить как старое, так и новое значения. Для этого задействуются два стека LIFO и станция переходит в ранг стековой (stacking station).

7.0.2.25



Алгоритм Token Ring. Приоритеты. Маркер не захвачен стековой станцией

7.0.2.26

За восстановление старого уровня приоритета в кольце ответственна стековая станция, осуществившая замену.

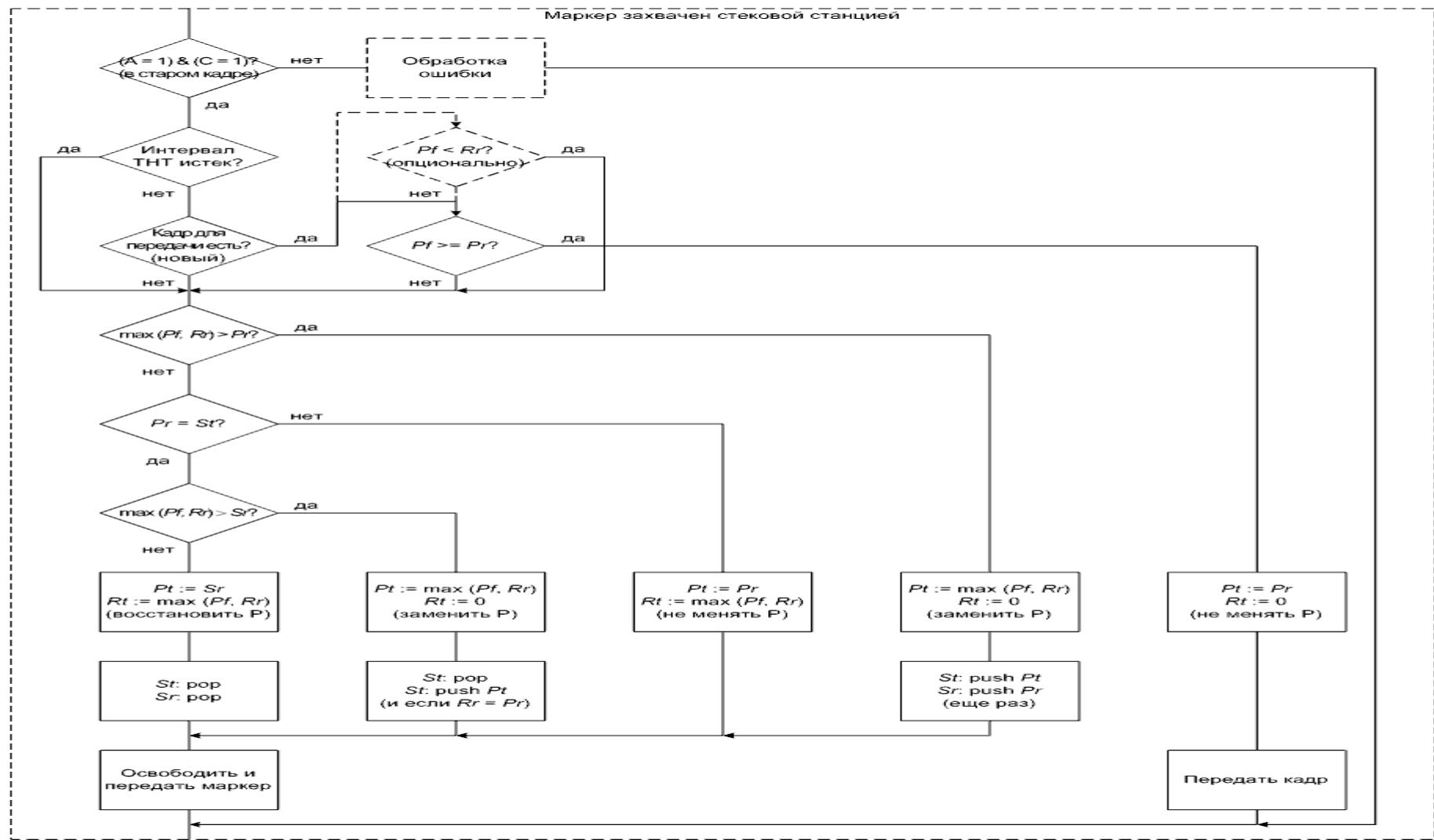
Попытка восстановления происходит при подходящих условиях. На стековой станции, не захватившей маркер, восстановление возможно только после принятия маркера -- когда обслужены все соответствующие запросы. При равенстве уровня приоритета в кольце уровню приоритета на вершине стека LIFO, в котором хранятся значения, на которые выполнялись замены, станция «понимает», что этот приоритет установила именно она.

Если при освобождении маркера нестековой станцией уровень приоритета в кольце может измениться только в большую сторону, то при обработке маркера стековой станцией только в меньшую сторону.

Приоритет в кольце не обязательно восстанавливается -- может быть понижен, опять же, до максимального из «представленных» уровней.

При освобождении маркера нестековой станцией приоритет в кольце не опускается именно для того, чтобы исключить «прыжки вниз», которые привели бы к бессмысленному восстановлению более высоких уровней.

7.0.2.27



Алгоритм Token Ring. Приоритеты. Маркер захвачен стековой станцией

7.0.2.28

Станция, захватившая маркер, вполне может быть стековой и, следовательно, сочетать в себе функции повышения и понижения приоритета в кольце.

При каждом захвате маркера и при каждой замене текущего уровня приоритета происходит сброс запрашиваемого уровня приоритета так как соответствующие запросы считаются учтеными.

Важно, что кадры на станции могут появляться произвольным образом, и это влияет на поведение.

Чтобы передать свой кадр станция не упускает случай. Неприоритетный кадр справедливо ждет своей очереди.

Чужой кадр станция никогда не буферизирует.

7.0.2.29

Раннее освобождение маркера совместимо с расширенной системой приоритетов. С одной стороны, эффективность использования кольца повышается, но, с другой стороны, приоритеты кадров учитываются хуже.

Соблюдение перечисленных выше правил гарантирует, что любая станция рано или поздно дождется возможности передать любой кадр.

7.0.2.30

Посредством кадров кроме сугубо пользовательской информации может передаваться и служебная, относящаяся к канальному уровню.

Отличительной особенностью Token Ring является то, что на канальном уровне регламентируется очень многое.

При этом в поле данных включаются так называемый вектор (vector) и некоторое количество подвекторов (subvectors).

Предусмотрено множество обязательных примитивов.

Временные интервалы контролируются большим количеством обязательных таймеров.

7.0.2.31

Кроме всего прочего, в Token Ring заложено несколько механизмов обеспечения надежности, включая автопереконфигурирование (autoreconfiguration) и сигнализацию об ошибках (beaconing).

Для предотвращения зацикливания станция-монитор метит каждый проходящий через нее кадр (маркер с $P > 0$) устанавливая значение бита M в единицу. Остальные станции этот бит не модифицируют. При исправном состоянии кольца уже помеченный кадр не должен еще раз «дойти» до станции-монитора. Если же это происходит, то станция-монитор инициирует «починку» кольца.

Во время передачи кадра при обнаружении ошибки станция прекращает передавать текущий кадр и передает прерывающую последовательность, тем самым сообщая принимающей станции о сбое в кадре.

Имеются возможности (в том числе аппаратные) гибкого подключения (inserting) и отключения (bypassing) станций от кольца.

7.0.2.32

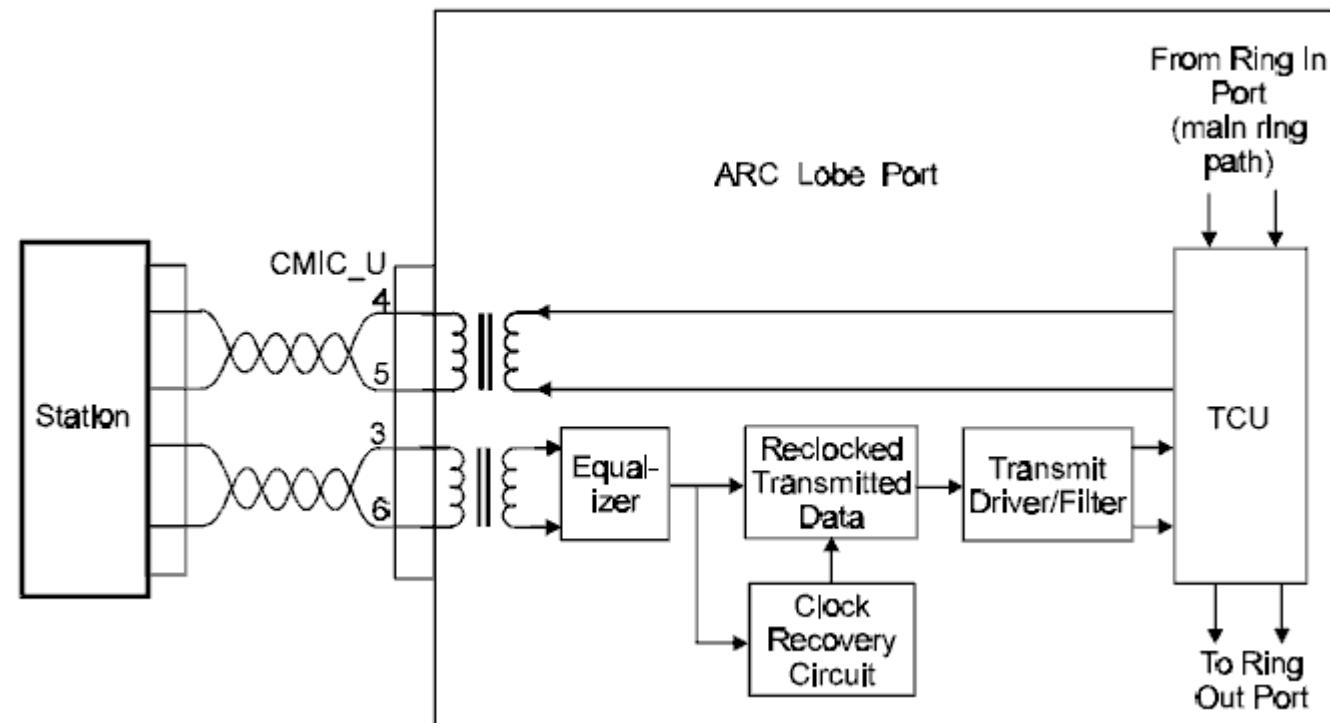
В качестве контрольного кода используется код CRC.

7.0.2.33

Скорость Token Ring равна 4 либо 16 Mbit/s (100 Mbit/s -- самые поздние реализации).

Предпринимали попытки разработать Wireless Token Ring.

7.0.2.34



Пример структуры активного концентратора Token Ring [IEEE]

7.0.3.1

Кроме Token Ring следует упомянуть еще ряд существующих технологий -- реализаций детерминированных методов.



Технология ARCNET (Attached Resource Computer NETwork) была первой технологией ЛКС, нашедшей массовое применение до экспансии Ethernet, в том числе благодаря своей дешевизне.

Стандарт ATA 878.1 был разработан и утвержден ARCNET Trade Association.

В настоящее время является сильно устаревшей.

Скорость: 2,5 Mbit/s.

Логическая топология: одностороннее кольцо.

Физическая топология: шина или звезда. Во втором случае требовалось дополнительное сетевое оборудование (пассивные или активные концентраторы).

Алгоритм являлся аналогом упрощенного варианта алгоритма Token Ring (без системы приоритетов).

7.0.3.2

Технологию Token Bus **разрабатывали** параллельно с Token Ring.

Была стандартизована как IEEE 802.4.

Благодаря плохому масштабированию и сложности восстановления после сбоев, почти не **применяли**, только в промышленные сети некоторых индустриальных компаний.

Разработка давно остановлена, является сильно устаревшей.

Скорость: 1, 5, 10, 20 Mbit/s.

Логическая топология: одностороннее кольцо.

Физическая топология: шина.

Алгоритм представлял собой адаптацию алгоритма Token Ring к шинной топологии.

7.0.3.3а

Технологию FDDI (Fiber Distributed Data Interface) разрабатывали целенаправленно для поддержки оптических СрПД, что позволяет значительно увеличить дальность передачи. Кроме собственно FDDI, еще был разработан аналогичный вариант для электрических СрПД под названием CDDI (Copper Distributed Data Interface).

FDDI формализовали в виде комплекса стандартов, которые разрабатывали постепенно -- в основном ANSI и ISO. Ключевыми являются стандарты: ISO 9314-1, ISO 9314-2 и ISO 9314-3.

FDDI был быстро вытеснен с рынка сетевых технологий после появления более дешевого Fast Ethernet, но ограниченно применяется до сих пор. CDDI распространения так и не получил.

Скорость: 100 Mbit/s, 200 Mbit/s.

7.0.3.3b

Логическая топология: однонаправленное кольцо с резервированием, то есть два отдельных кольца (если оба кольца исправны, то они функционируют параллельно).

Физическая топология: двойное кольцо, к которому с помощью дополнительного сетевого оборудования могут подключаться деревья (узлами дерева являются концентраторы, листьями -- станции, концентратор-корень **включают** в двойное кольцо).

Алгоритм представляет собой расширение алгоритма Token Bus.

7.0.3.4

Технология 100VG-AnyLAN была разработана HP и стала альтернативой Fast Ethernet. Идея заключалась в получении по тем временам высокоскоростного гибрида между Ethernet и Token Ring, причем с сохранением совместимости с их кадрами.

Позже была стандартизована как IEEE 802.12.

На технологию **возлагали** большие надежды, но она была быстро отвергнута рынком и в скорости практически исчезла.

Скорость: 100 Mbit/s.

Логическая топология: дерево.

Физическая топология: дерево (с опциональным резервированием), формируемое с помощью дополнительного сетевого оборудования (узлами дерева являются повторители, листьями -- станции или мосты, с помощью мостов можно подключать сегменты Ethernet или Token Ring).

Метод доступа получил название Demand-priority. **Основан** на программном автомате под названием MAC state machine.

7.0.4.1

Таким образом, существуют три основных способа выбора активного передатчика:

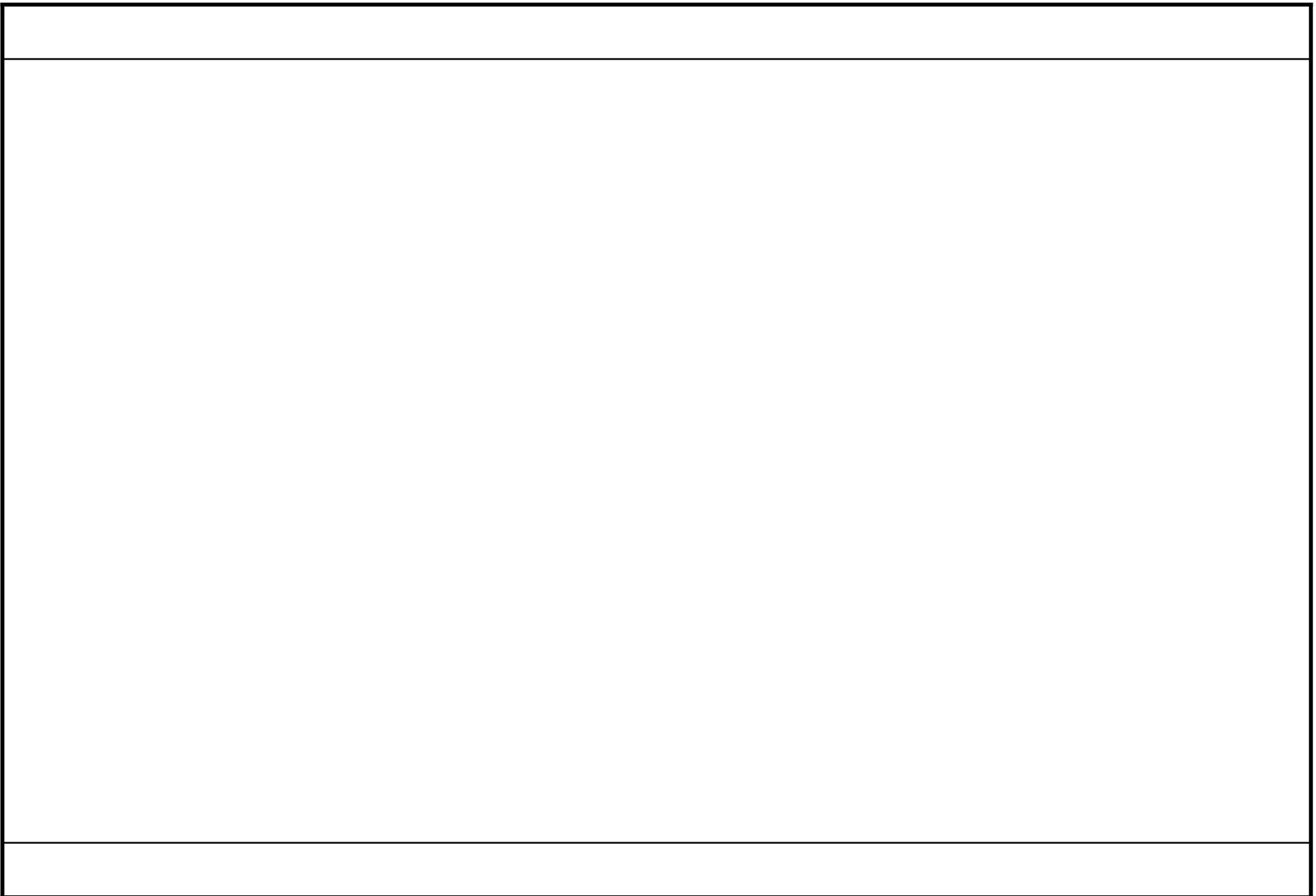
1. Перепасовка маркера (token passing).
2. Резервирование (reservation).
3. Опрос (polling).

Выбор может происходить и по расписанию.

7.0.5.1

Если в топологиях с множественным доступом канал не является моноканалом, то совместно использоваться он может следующими методами (channelization):

1. FDMA (Frequency Division Multiple Access) -- множественный доступ на разных частотах (частотное разделение).
2. TDMA (Time Division Multiple Access) -- множественный доступ на одной частоте в разные временные окна (временноное разделение).
3. CDMA (Code Division Multiple Access) -- множественный доступ на одной частоте с изменением параметров кодирования.



АДРЕСАЦИЯ В КОМПЬЮТЕРНЫХ СЕТЯХ

8.0.1.1

Для того, чтобы станции-абоненты могли организовать взаимодействие, им необходимо некоторым образом выделять друг друга среди других станций.

С целью идентификации станций им присваивают некоторые адреса. Таким образом, возникает *адресация (addressing)* в СПД.

8.0.1.2а

Как было сказано ранее, в форматах большинства пакетов присутствуют два адреса:

1. Адрес назначения (destination address).
2. Адрес источника (source address).

В процессе пересылки пакета между абонентами адресация играет ключевое значение.

Производительность СПД напрямую зависит от расположения адресов в пакете. Поэтому адреса «выносят» в самое начало пакета. Более того, поскольку с точки зрения доставки пакета адрес назначения является более важным (в СПД анализируется именно этот адрес), он как правило располагается раньше.

8.0.1.2b

Многие топологии предполагают возможность приема переданного одной из станций пакета всеми остальными станциями в пределах сегмента -- вне зависимости от того, какой из станций пакет был предназначен.

Следует различать действия «принят станцией», «проанализирован станцией» и «обработан станцией». Факт приема станцией пакета подразумевает, что пакет будет проанализирован, но не подразумевает «полноценную» обработку. Именно сравнение считанного из принятого пакета адреса назначения со своим адресом, позволяет станции распознать пакет как «свой».

Считанный из пакета адрес источника позволяет станции (при необходимости) определить абонента, создавшего пакет.

8.0.2.1

Следует учитывать, что важное влияние на адресацию оказывает инкапсуляция. Адресация всегда «привязана» к некоторому протоколу, а протокол, в свою очередь, «привязан» к уровню модели OSI. Поэтому закономерно, что на каждом из уровней присутствует своя независимая система адресации.

Пакет, воспринятый как «свой» на одном из уровней, после его передачи на более высокий уровень, там вполне может быть «отвергнут». Кроме того, «окончательная» обработка не всегда происходит на прикладном уровне (классический пример: ретрансляция пакета между сегментами при маршрутизации).

8.0.2.2

В каждом пакете должны присутствовать по крайней мере адреса канального уровня.

В большинстве же практических реализаций семейств протоколов, кроме адресации на канальном уровне, предусмотрена адресация на сетевом (в связке с транспортным) и прикладном уровнях.

Допустимость повторения адресов на одном уровне вытекает из цели разработки определенного протокола.

8.0.2.3

Адреса канального уровня «зашиваются» в сетевое оборудование при его производстве и поэтому повторяться не должны. Они не предполагают возможность пользовательского вмешательства и их считают абсолютно уникальными. Часто (в том числе Cisco) такую адресацию называют **физическими** (*physical*).

Адреса сетевого и прикладного уровней назначают пользователи. Часто (в том числе Cisco) такую адресацию называют **логическими** (*logical*).

8.0.2.4

В нормальной ситуации, по крайней мере в течение сеанса взаимодействия, адреса разных уровней одной станции должны соответствовать друг другу. Поэтому возникает необходимость в служебных протоколах, отыскивающих эти соответствия.

8.0.2.5

Кроме всего прочего, даже на одном уровне модели OSI адресация может быть *иерархической* (hierarchical), то есть предполагать определенную структуризацию соответствующего адресного пространства.

Иерархичность выражается в количественном и качественном разделении адресов на типы.

8.0.2.6

Одним из примеров может служить связка адреса сетевого уровня с адресом транспортного уровня.

В рамках функционирования сетевой ОС можно выделить объекты:

1. *Сетевой процесс* (network process) -- представляет собой пару: процессор и выполняющаяся на нем сетевая (то есть использующая сетевые ресурсы) программа; причем, если меняется хотя бы один из этих компонентов, то получается новый процесс.

2. *Сетевой ресурс* (network resource) -- это любой компонент вычислительной системы, который может быть предоставлен в пользование сетевому процессу на определенное время.

Для того, чтобы взаимодействующие сетевые процессы могли найти друг друга, во всех реальных системах используется три уровня адресации:

1. Необходимо адресовать подсеть -- используется *адрес подсети* (subnet address).

2. Необходимо адресовать станцию в подсети -- используется *адрес станции* (node address).

3. Необходимо адресовать процесс в станции -- используется так называемый *адрес программного порта* (software port).

8.0.2.7

Под адрес порта, как правило, отведено два байта.

При назначении программных портов учитываются диапазоны, к которым они относятся.

Диапазоны программных портов применительно к семейству TCP/IP.

Port Number Range	Port Group
0 – 1023	Well Known
1024 – 49151	Registered
49152 – 65535	Private and Dynamic

Так называемые хорошо известные порты предназначены для адресации основных сервисов в Internet.

Порты для дополнительных публичных сервисов нужно регистрировать.

Порты для приватных (и редких) сервисов регистрировать не нужно.

8.0.2.8

Для чего нужны динамические порты?

8.0.3.1

Специально для компьютерных сетей были разработаны четыре основных способа адресации, которые заключаются в применении адресов четырех базовых типов:

1. *Юникаст* (unicast) -- пакет с таковым адресом назначения должен быть обработан одной соответствующей станцией.
2. *Бродкаст* или, по-другому, *широковещательных* (broadcast) -- пакет с таковым адресом назначения должен быть обработан всеми станциями.
3. *Мультикаст* (multicast) -- пакет с таковым адресом назначения должен быть обработан несколькими станциями из множества.
4. *Эникаст* (anycast) -- пакет с таковым адресом назначения должен быть обработан одной станцией из множества.

По сути, мультикаст- и эникаст-адреса являются *групповыми идентификаторами* (group IDs).

8.0.3.2

Специфика тех или иных типов накладывает ограничения на возможность использования адресов.

Бродкаст-, мультикаст- и эникаст-адреса не могут быть адресами источников, так как отдельно взятый пакет может сгенерировать только одна станция.

8.0.3.3

Особую проблему представляет собой межсегментная ретрансляция группового трафика (актуально для прикладного мультикаст-трафика).

Проблема решается с помощью дополнительных служебных протоколов.

8.0.3.4

Наиболее сложной формой адресации является эникаст-адресация. Очевидно, что каждый раз при приеме эникаст-пакета должен осуществляться выбор на основе какого-либо критерия.

При этом адресуемые станции должны осуществлять выбор в пределах группы сами.

Отправившая пакет станция не может принимать участие в алгоритме выбора, она уже сделала свой «выбор» записав в пакет в качестве адреса назначения эникаст-адрес.

Выбор должен быть сделан заранее, чтобы принимающая станция была готова к поступлению в группу пакета.

Примером критерия выбора может служить время задержки.

Выбор может осуществляться однократно либо периодически.

8.0.3.5

Придумайте алгоритм децентрализованного выбора в группе, если станции заранее ничего друг о друге не знают.

8.0.4.1

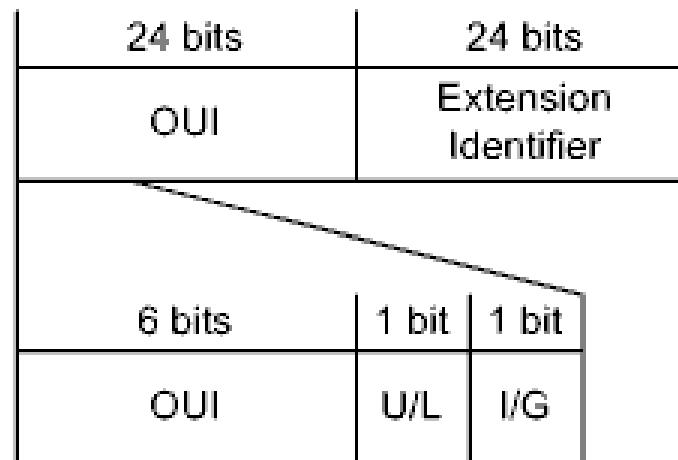
Почти все широко применяемые в настоящее время технологии ЛКС (например, Ethernet) разработаны IEEE, поэтому производители соответствующего сетевого оборудования соблюдают правила, сформулированные в стандартах этой организации.

Уникальность MAC-адресов контролирует IEEE RA (IEEE Registration Authority).

В стандартах IEEE определены три базовых формата MAC-адресов: MAC-48, EUI-48 и EUI-64, где EUI (Extended Unique Identifier) -- расширенный уникальный идентификатор.

MAC-48 можно считать синонимом EUI-48, хотя изначально EUI-48 было более общим понятием.

8.0.4.2



00-16-41-57-7D-48

Поля:

OUI (Organizationally Unique Identifier) -- уникальный идентификатор организации (производителя).

U/L (Universal/Local) -- признак универсальности-локальности адреса.

I/G (Individual/Group) -- признак индивидуального-группового адреса.

Extension Identifier -- идентификатор-наполнитель.

Формат EUI-48

8.0.4.3

OUIs выдают централизовано, уникальность оставшейся части должны обеспечивать сами организации (любым способом по своему усмотрению).

Время валидности адресов (время, которое нужно выдержать перед повторным присвоением того же адреса другому устройству) определено как 100 лет.

8.0.4.4

Иногда, при администрировании, возникает необходимость подменить адрес, «зашитый» в оборудование, на некий другой.

Этот новый адрес называют *локальным административным адресом* (locally administered address).

Его признаком является единичное значение бита U/L.

Согласовывать значение остальных битов не требуется, но в пределах сегмента адрес не должен повторяться.

8.0.4.5

Граница между OUI и Extension Identifier может проходить не только посередине адреса. В общем случае предусмотрены три варианта разрядности поля OUI:

1. MA-L (MAC Address -- Large) -- 24 бита (данную схему использовали до 1 января 2014 г.)
2. MA-M (MAC Address -- Medium) -- 28 битов (схема доступна после 1 января 2014 г.)
3. MA-S (MAC Address -- Small) -- 36 битов (схема доступна после 1 января 2014 г.)

Иногда поле OUI рассматривают как CID (Company ID), что, по большому счету, то же самое -- зависит от комбинации значений битов U/L и I/G (рассматривают уже как биты X и M соответственно).

8.0.4.6

MAC-адреса вполне могут «фигурировать» вне адресных полей кадра (например, в поле данных служебных кадров).

При формировании кадров Ethernet и Wi-Fi, во всех полях, кроме FCS, младшие биты отдельно взятых байтов располагаются ближе к началу кадра («задом наперед» в сравнении с рисунком).

При последующем сдвиге в канал порядок битов не изменяется.

Таковое представление MAC-адресов принято называть каноническим (canonical).

А вот при формировании кадров Token Ring, во всех полях, кроме как раз DA и SA, младшие биты отдельно взятых байтов располагаются дальше от начала кадра (как на рисунке).

Поэтому в разнородных СПД могут возникать проблемы из-за неправильной интерпретации MAC-адресов.

8.0.4.7

По правилам IEEE MAC-адреса записывают в следующей нотации:

XX-XX-XX-XX-XX-XX

Где X -- шестнадцатеричная цифра (верхний регистр).

Но очень часто используют альтернативные нотации.

Примеры:

00-16-41-57-7D-48 -- IEEE

00-16-41-57-7d-48

00:16:41:57:7D:48

00:16:41:57:7d:48

0016.4157.7d48 -- Cisco

8.0.4.8

Все юникаст-MAC-адреса должны иметь нулевое значение бита I/G.

Групповые MAC-адреса формируются по особым правилам, которые будут рассмотрены позже.

В качестве бродкаст-MAC-адреса принято использовать значение FF-FF-FF-FF-FF-FF.

8.0.4.9

Следует отметить, что EUI-64 может использоваться не только для адресации, а и для просто идентификации устройств.

Примеры технологий с применением EUI-48: Ethernet, Wi-Fi, Token Ring.

Примеры технологий с применением EUI-64: IPv6, FireWire.

8.0.5.1a

В семействе TCP/IP за адресацию на сетевом уровне отвечает протокол IP.

Заголовок протокола IPv4 (версии 4) (RFC 791) имеет фиксированную структуру.

octet	octet	octet	octet		
Version	IHL	Type of Service	Total Length		
Identification		Flags	Fragment Offset		
Time to Live	Protocol	Header Checksum			
Source Address					
Destination Address					
Options		Padding			

Формат заголовка IPv4

8.0.5.1b

Поля:

Version -- версия (значение равно 4).

IHL (Internet Header Length) -- длина заголовка (в 32-ухбитных словах, минимальное значение равно 5).

Type of Service -- тип сервиса (связано с QoS).

Total Length -- общая длина данных (в байтах, не может превышать 65535 байтов).

Identification -- уникальный идентификатор пакета (при фрагментации позволяет определить к какому пакету относится фрагмент).

Flags -- флаги.

Fragment Offset -- смещение текущего фрагмента (в 64-ехбитных словах, смещение первого фрагмента равно нулю).

Time to Live -- «время жизни» (при каждой ретрансляции уменьшается, когда становится равным нулю пакет уничтожается).

Protocol -- протокол (инкапсулируемый в поле данных).

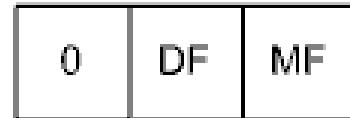
Header Checksum -- контрольная сумма заголовка.

Source Address -- адрес источника.

Destination Address -- адрес назначения.

Options -- опции (например, связанные с безопасностью, размер вариативен).

8.0.5.2



Флаги:

DF (Don't Fragment): 0 -- пакет фрагментирован, 1 -- пакет нефрагментирован.

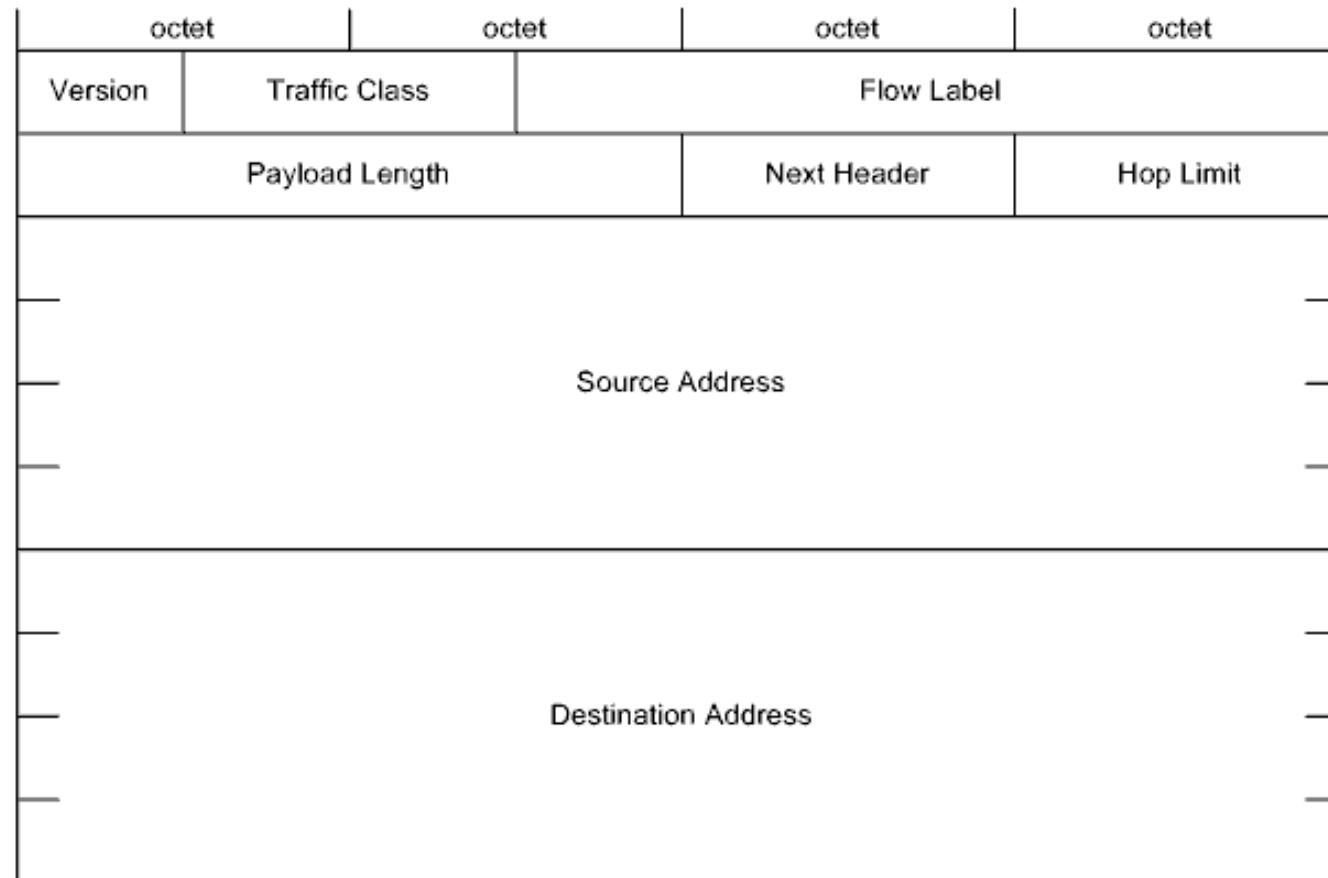
MF (More Fragments): 0 -- текущий фрагмент является последним, 1 -- текущий фрагмент не является последним.

Поле Flags

8.0.5.3а

Заголовок протокола IPv6 (RFC 8200) имеет «гибкую» структуру.

Заголовки «каскадируются» -- сколько заголовков нужно, столько и вставляется.



Формат заголовка IPv6

8.0.5.3b

Новые поля:

Traffic Class -- класс трафика (связано с QoS).

Flow Label -- метка потока (связано с QoS).

Payload Length -- длина полезной нагрузки (в байтах, аналог поля Total Length).

Next Header -- селектор следующего заголовка (в том числе, аналог поля Protocol).

Hop Limit -- ограничитель числа «прыжков» (аналог поля Time to Live).

8.0.5.4

Полноценная реализация IPv6 должна поддерживать следующие заголовки:

1. IPv6 header -- собственно IPv6-заголовок.
2. Hop-by-Hop Options header -- заголовок опций ретрансляции.
3. Destination Options header -- заголовок предназначенных станции назначения опций.
4. Routing header -- маршрутизационный заголовок.
5. Fragment header -- заголовок фрагмента.
6. Authentication header -- заголовок протокола AH (связано с защитой информации).
7. Encapsulating Security Payload header -- заголовок протокола ESP (связано с защитой информации).
- +8. Upper-layer header -- заголовок протокола вышестоящего уровня.

Подробно IPv4- и IPv6-адресация будет рассмотрена в дальнейшем.

8.0.5.5

Охарактеризуйте протокол IP, исходя из уже полученных знаний.

8.0.6.1

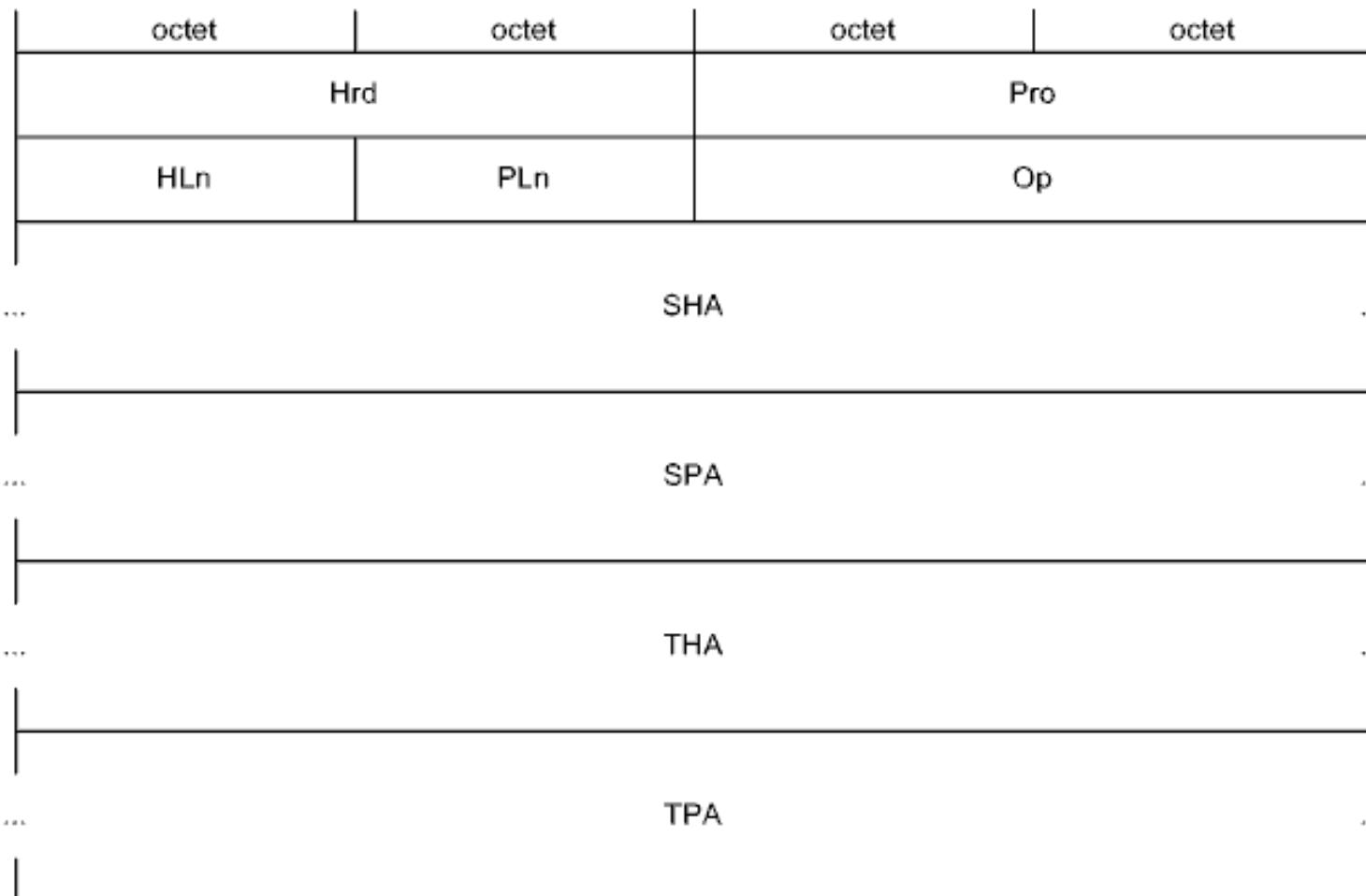
Группа протоколов под названием ARPs (Address Resolution Protocols) предназначена для восстановления соответствий между MAC-адресами и IP-адресами.

Под прямым преобразованием, собственно ARP (RFC 826), понимают нахождение MAC-адреса по IP-адресу.

Обратное преобразование выполняется по протоколу RARP (Reverse ARP).

Существует еще InARP (Inverse ARP) и некоторые другие расширения, которые, как и практическое применение ARP, будут рассмотрены в дальнейшем.

8.0.6.2a



Формат пакета ARP

8.0.6.2b

Поля:

Hrd (Hardware) -- тип оборудования (1 -- Ethernet).

Pro (Protocol) -- протокол (800h -- IP).

HLn (Hardware address Length) -- длина аппаратного (физического) адреса (в байтах, 6 -- Ethernet).

PLn (Protocol address Length) -- длина протокольного (логического) адреса (в байтах, 4 -- IP).

5. Op (Opcode) -- код операции: 1 -- Request -- запрос, 2 -- Reply -- ответ (и некоторые другие).

6. SHA (Sender Hardware Address) -- аппаратный адрес станции-отправителя (запрашивающей либо отвечающей на запрос).

7. SPA (Sender Protocol Address) -- протокольный адрес станции-отправителя.

8. THA (Target Hardware Address) -- аппаратный адрес станции-получателя.

9. TPA (Target Protocol Address) -- протокольный адрес станции-получателя.

8.0.7.1

Протокол системы DNS (Domain Name System) (два основных RFCs, RFC 1034 -- больше теория, RFC 1035 -- больше практика) предназначен для восстановления соответствий между IP-адресами и адресами прикладного уровня.

8.0.7.2

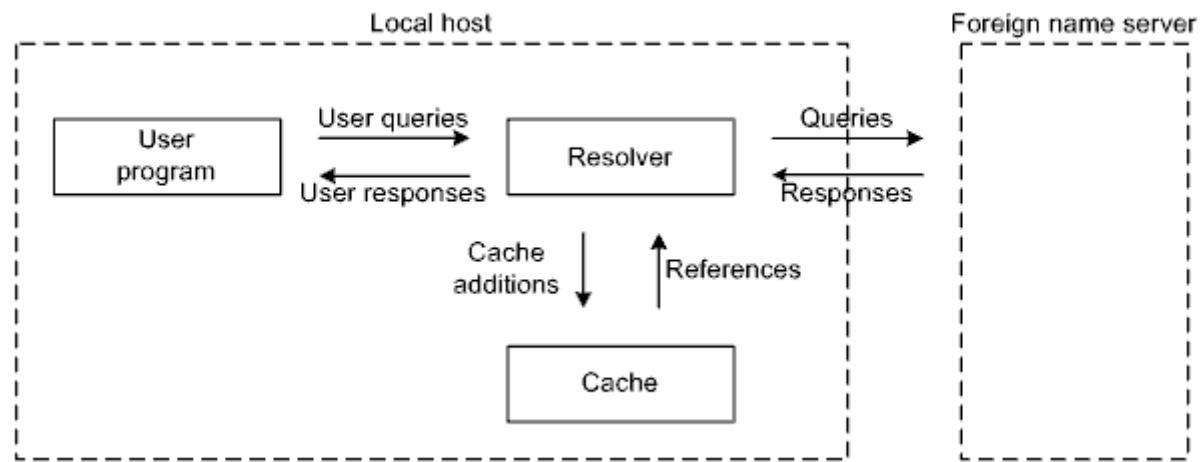
Следует отметить, что под *доменом* (domain, иногда cloud) в СПД обобщенно понимают совокупность устройств, работающих в рамках некоторых единых правил.

8.0.7.3

Некоторые служебные протоколы, в том числе DNS, нельзя однозначно сопоставить с моделью OSI.

Исходя из инкапсуляции, протокол DNS следует условно отнести к прикладным.

8.0.7.3



Структура системы DNS

8.0.7.4

Система DNS соответствует клиент-серверной модели и включает три основных компонента:

1. Адресное пространство доменных названий (domain name space) и записи о ресурсах -- RRs (Resource Records).
2. Серверы названий (name servers).
3. Программы, отвечающие на запросы клиентов (resolvers).

Каждый из этих компонентов «видит» систему DNS по-своему.

8.0.7.5а

Адресное пространство доменных названий имеет иерархическую древовидную структуру.

Каждый узел дерева на некотором уровне иерархии обозначают *DNS-меткой* (*DNS label*) длиной от 0 до 63 байтов (должна начинаться с буквы и состоять из комбинации букв любого регистра, цифр и символа -). Метка нулевой длины зарезервирована и является корнем дерева. При присоединении станции к определенному домену ей так же присваивают метку.

Доменное название строится из меток -- в соответствии с путем к корневой метке. Полная длина не может превышать 255 байтов.

Доменное название может относиться как к отдельно взятой станции, так и к некоторой ветви дерева, то есть к *DNS-домену* (*DNS domain*).

Доменное название может быть как абсолютным (*absolute*), то есть содержащим всю цепочку меток от станции до корневой метки, так и относительным (*relative*), то есть содержащим только часть меток.

Внутреннее представление метки: один байт, в котором указана длина метки, за которым следуют собственно байты метки. При интерпретации меток регистр букв не учитывается.

8.0.7.5b

Согласно принятой нотации записи доменных названий метки разделяют точками и корневая метка является крайней справа.

8.0.7.6

Напишите пример цепи из доменных названий.

8.0.7.7

Изначально, когда сеть Internet была сосредоточена на территории США, базовым критерием структуризации доменных названий Internet-сайтов являлось целевое использование. Были зарегистрированы следующие домены первого уровня -- TLDs (Top Level Domains): .arpa (ARPANET), .com (commerce), .edu (education), .gov (government), .int (international), .mil (military), .net (network), и .org (organization).

В дальнейшем, по мере расширения Internet, широкое распространение получили национальные TLDs, например, .BY.

С недавнего времени основной упор сделан на продвижение национальных языков (в качестве альтернативы английскому языку). Зарегистрированы дополнительные национальные TLDs, например, .БЕЛ.

Четыре TLDs зарезервированы для специального использования: .example, .invalid, .localhost, .test.

8.0.7.8

Серверы названий удерживают БД с записями о ресурсах.

Серверы названий делят на:

1. *Авторитетные* (authoritative, master) -- являются первоисточниками информации о некоторых частях системы DNS, называемых зонами (zones).
2. *Вспомогательные* (non-authoritative, slave) -- работающие на основании сведений от авторитетных серверов.

Таким образом, серверы так же образуют иерархию -- вплоть до наличия корневых серверов.

8.0.7.9

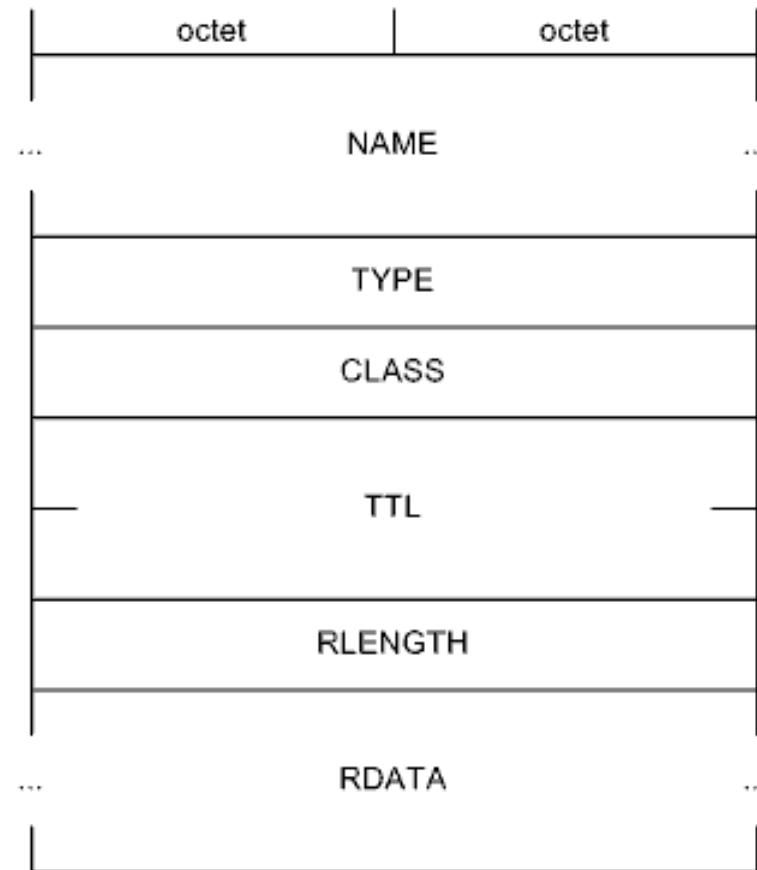
Под прямым преобразованием понимают нахождение IP-адреса по доменному названию.

Возможно и обратное преобразование.

8.0.7.10

Каждой входящей в систему DNS станции (как и каждому домену) соответствует некоторое количество RRs.

8.0.7.11a



Формат DNS RR

8.0.7.11b

Поля:

1. NAME -- доменное название (к которому относится RR, целевое при поиске).
2. TYPE -- тип.
3. CLASS -- класс (семейство протоколов).
4. TTL (Time To Live) -- «время жизни» (то есть время валидности RR, в секундах).
5. RLENGTH (Resource LENGTH) -- длина данных ресурса.
6. RDATA (Resource DATA) -- данные ресурса (зависят от типа и класса).

8.0.7.12а

Основные типы RR:

1. A (A host address) -- IP-адрес хоста.
2. NS (Name Server) -- авторитетный сервер названий домена.
5. CNAME (Canonical NAME) -- каноническое доменное название (станции либо домена, для псевдонима).
6. SOA (Start Of a zone of Authority) -- оригинальные параметры зоны (сервер с изначальным описанием зоны, контактное лицо, время валидности и другие).
10. NULL -- нулевая запись (произвольная информация).
12. PTR (PoinTeR) -- указатель -- доменное название станции (при обратных преобразованиях).
13. HINFO (Host INFO) -- информация о станции (процессор и ОС).
15. MX (Mail eXchange) -- доменное название почтового сервера в домене (включая приоритет, этот тип используется и вместо нескольких отмененных типов).
16. TXT (TeXT strings) -- текстовые строки (либо строка).
28. AAAA (--) -- IPv6-адрес хоста (RFC 3596).

8.0.7.12b

33. SRV (SeRVer selection) -- описание сервиса (любого дополнительного сетевого сервиса на станции, например, файлового) (RFC 2782).

И некоторые другие.

8.0.7.13

Классы RR:

1. IN -- Internet.
2. CS -- CSNET (устарел и аннулирован).
3. CH -- Chaosnet (устарел).
4. HS -- Hesiod (для БД, очень редкий).

Остальные значения классов зарезервированы.

8.0.7.14

Примеры значений RRs класса IN:

A: 192.168.11.1.

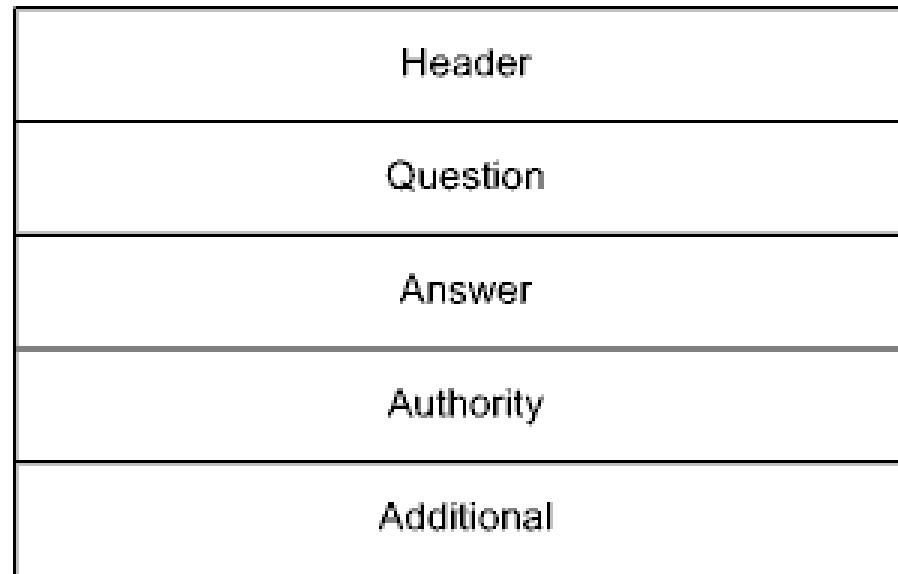
CNAME: 5-508-fileserv.bsuir.by.

MX: 10 mail.bsuir.by.

NS: proxy1.bsuir.by.

PTR: 5-508-fileserv.bsuir.by.

8.0.7.15



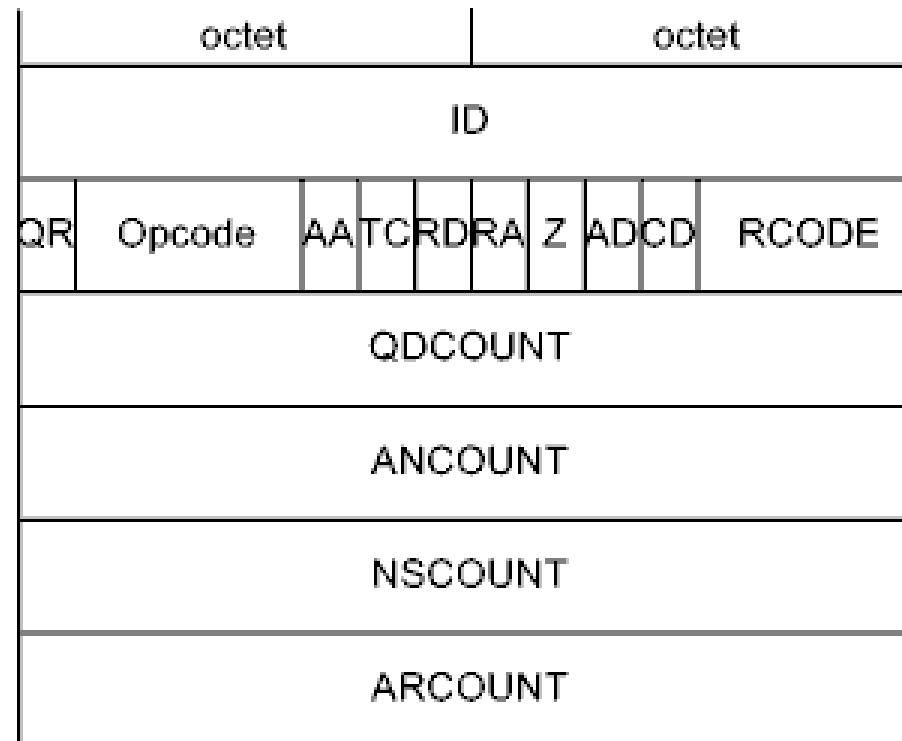
Поля:

1. Header -- заголовок.
2. Question -- вопрос.
3. Answer -- ответ.
4. Authority – авторитетный ответ.
5. Additional -- дополнение.

Заголовок присутствует всегда, остальные поля вариативны.

Формат сообщения DNS

8.0.7.16a



Формат заголовка сообщения DNS

8.0.7.16b

Поля:

1. ID (IDentifier) -- идентификатор (программы, сгенерировавшей запрос).
2. QR (Query/Responce) -- флаг запроса-ответа: 0 -- Query -- запрос, 1 -- Responce -- ответ.
3. OPCODE (OPeration CODE) -- код операции (запроса): 0 -- QUERY (standard QUERY) -- стандартный запрос (о прямом преобразовании), 1 -- IQUERY (Inverse QUERY) -- запрос об обратном преобразовании (RFC 3425 отменен, альтернатива -- использование PTR RR), 2 -- STATUS (server STATUS request) -- запрос состояния сервера, 4 -- NOTIFY -- уведомление (об изменениях в БД о зоне) (RFC 1996), 5 -- UPDATE -- обновление (динамическое обновление БД о зоне) (RFC 2136), 6 -- DSO (DNS Stateful Operations) -- стабильные DNS-операции (альтернативный унифицированный синтаксис) (RFC 8490), остальные значения зарезервированы.
4. AA (Authoritative Answer) -- флаг авторитетного ответа.
5. TC (TrunCation) -- флаг «усеченности» сообщения (при слишком длинном сообщении).

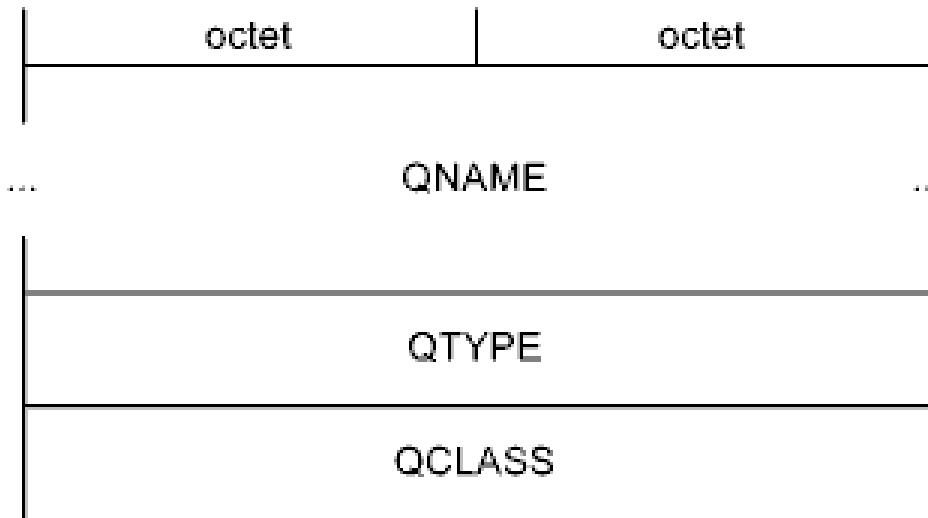
8.0.7.16c

6. RD (Recursion Desired) -- флаг желательной рекурсии (при обработке запроса).
7. RA (Recursion Available) -- флаг поддержки рекурсии.
8. Z (Zero) -- нулевой бит (зарезервировано).
9. AD (Authenticated Data) -- флаг криптографической верификации ответа (RFC 4035).
10. CD (Checking Disabled) -- флаг отсутствия необходимости в криптографической верификации ответа (при запросе) (RFC 4035).
11. RCODE (Response CODE) -- код ответа: 0 -- NoError (No Error) -- ошибок нет, 1 -- FormErr (Format Error) -- ошибка в формате, 2 -- ServFail (Server Failure) -- сбой сервера, 3 -- NXDomain (Non-eXistent Domain Name) -- доменное название не существует, 4 -- NotImp (Not Implemented) -- запрос не поддерживается, 5 -- Refused (Query Refused) -- запрос отклонен, остальные значения относятся к расширениям DNS (RFC 2136, RFC 2845, RFC 2930, RFC 4635, RFC 6672, RFC 6891, RFC 7873, RFC 8490) и зарезервированы.

8.0.7.16d

12. QDCOUNT (Query DNS COUNT) -- количество элементов (RRs) в поле Question (обычно один).
13. ANCOUNT (ANswer COUNT) -- количество элементов (RRs) в поле Answer.
14. NSCOUNT (Name Server COUNT) -- количество элементов (RRs) в поле Authority.
15. ARCOUNT (Additional Records COUNT) -- количество элементов (RRs) в поле Additional.

8.0.7.17



Поля:

1. QNAME (Query NAME) -- доменное название в запросе.
2. QTYPE -- (Query TYPE) -- тип запроса.
3. QCLASS (Query CLASS) -- класс запроса.

Формат элемента поля Question

8.0.7.18

Множество значений QTYPE является расширением множества значений TYPE. Основные из новых типов:

251. IXFR (Incremental zone i.e. X transFeR) -- запрос текущих изменений в БД о зоне (от вспомогательного сервера авторитетному, по одноименному протоколу) (RFC 1995).

252. AXFR (Authoritative zone i.e. X transFeR) -- запрос полной БД о зоне (по одноименному протоколу) (+RFC 5936).

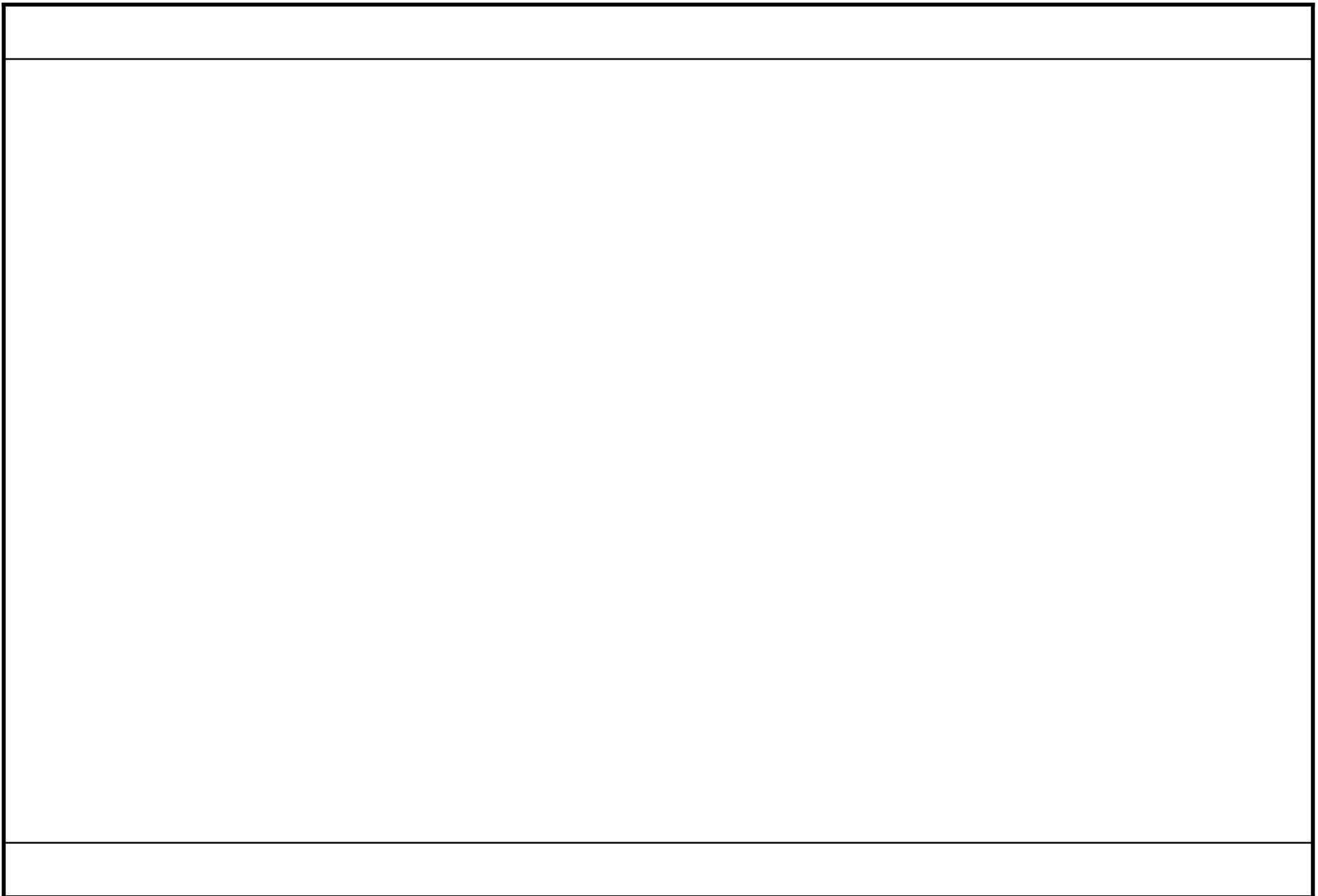
255. * -- запрос всех RRs.

Множество значений QCLASS является расширением множества значений CLASS. Новый класс:

255. * -- любой класс.

8.0.7.19

Практическое применение DNS будет рассмотрено в дальнейшем.



МЕТОДЫ ВЗАИМОДЕЙСТВИЯ В ЗВЕНЕ СЕТИ ПЕРЕДАЧИ ДАННЫХ

9.0.1.1

При формировании полномасштабной сети, то есть при объединении разрозненных физических сегментов в СПД той или иной сложности, возникает ряд специфических задач, направленных на оптимизацию взаимодействия между абонентами.

Упомянутые задачи решают на третьем и четвертом уровнях модели OSI.

Новые задачи обусловлены серьезными различиями процессов передачи-приема пакета в пределах сегмента и между сегментами. Основные отличия заключаются в необходимости ретрансляций пакетов, а так же в возможном наличии альтернативных путей.

9.0.1.2

Одним из ключевых терминов транспортного уровня является термин *соединение* (*connection*). По сути дела, понятие соединения связано с понятием готовности. Если абоненты находятся в состоянии «нормальной готовности» передавать или принимать данные, то считают, что между ними установлено соединение.

С учетом абстрагирования от более низких уровней модели OSI и инкапсуляции, соединение может быть выражено неявно.

9.0.1.3

Нужно отличать *виртуальные соединения* (virtual connections) от *физических соединений* (physical connections).

Абоненты-программы физически (явно) соединены быть не могут. Следовательно, применительно к ним, соединения являются сугубо виртуальными.

9.0.1.4

Следует также учитывать, что нормальная готовность может рассматриваться в двух ракурсах:

1. Организация взаимодействия абонентов-программ.

2. Настройка задействованного промежуточного оборудования.

В первом случае речь идет о собственно виртуальных соединениях транспортного уровня, во втором -- о *виртуальных цепях* (virtual circuits) сетевого или канального уровней.

В свою очередь, виртуальные цепи бывают:

1. PVCs (Permanent Virtual Circuits) -- *выделенные виртуальные цепи*.

2. SVCs (Switched Virtual Circuits) -- *коммутируемые виртуальные цепи* (в отечественной литературе иногда называют *виртуальными вызовами*).

Термин *виртуальный канал* (virtual channel) может в равной степени подходить как к виртуальным соединениям, так и к виртуальным цепям.

9.0.1.5

При разговоре о соединениях невозможно обойти стороной вопрос о надежности.

Существуют два способа организации взаимодействия:

1. Без гарантированной доставки -- в СПД предпринимаются определенные усилия по доставке пакетов, но при этом ничего не гарантируется (при необходимости, соответствующий контроль возлагается на программы-абоненты).

2. С гарантированной доставкой -- алгоритм работы транспортной службы гарантирует доставку пакетов (программы-абоненты могут не контролировать наличие и очередность пакетов).

Однако, соединение без гарантированной доставки практического смысла не имеет. Поэтому наличие соединения как правило говорит о надежности.

9.0.1.6

В общем случае, контроль передачи информации посредством СПД предотвращает не только потерю пакетов, но и искажение их содержимого.

Отсутствие соединения не означает, что защита от сбойных пакетов отсутствует.

9.0.2.1

Простейшим подходом к обеспечению контроля доставки информационных пакетов является применение метода, который обобщенно можно назвать методом запросов-подтверждений (*requests/acknowledges*).

Метод предполагает некоторое разнообразие и заключается в том, что вводят специальные служебные пакеты двух типов.

Пакет-запрос используется при получении права принять или передать полезные данные, а также собственно при запросе данных.

Пакет-подтверждение (в отечественной литературе часто называют *квитанцией*) передается в ответ на пакет-запрос или после приема полезных данных.

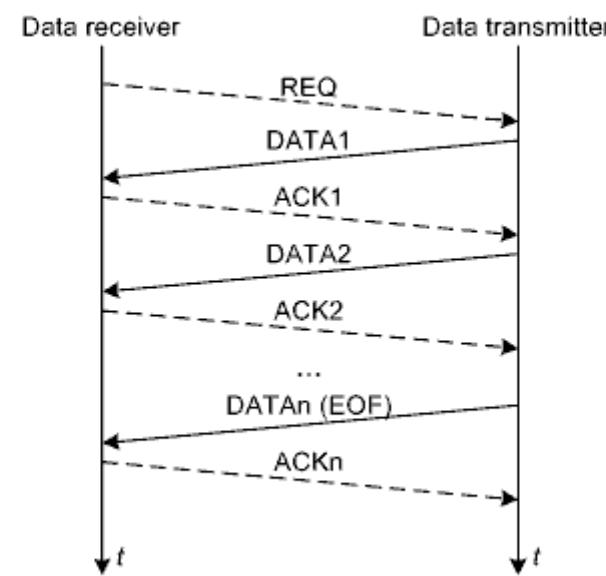
9.0.2.2

Кроме того, при реализации метода запросов-подтверждений следует учитывать следующие обстоятельства:

- инициатором взаимодействия может быть передатчик либо приемник информационных пакетов;
- контроль может осуществляться передатчиком либо приемником, либо передатчиком и приемником совместно;
- запросы либо подтверждения могут отсутствовать вообще;
- запросы могут комбинироваться с подтверждениями;
- запрашиваться и подтверждаться может все сообщение либо каждый из пакетов;
- подтверждаться могут не только информационные, а и служебные пакеты;
- квитанции могут быть как положительными, так и отрицательными;
- факт потери пакета может определяться и обрабатываться по-разному.

Таким образом, не смотря на сохранение идеологии, практические реализации метода запросов-подтверждений могут сильно различаться.

9.0.2.3



Пример взаимодействия методом запросов-подтверждений

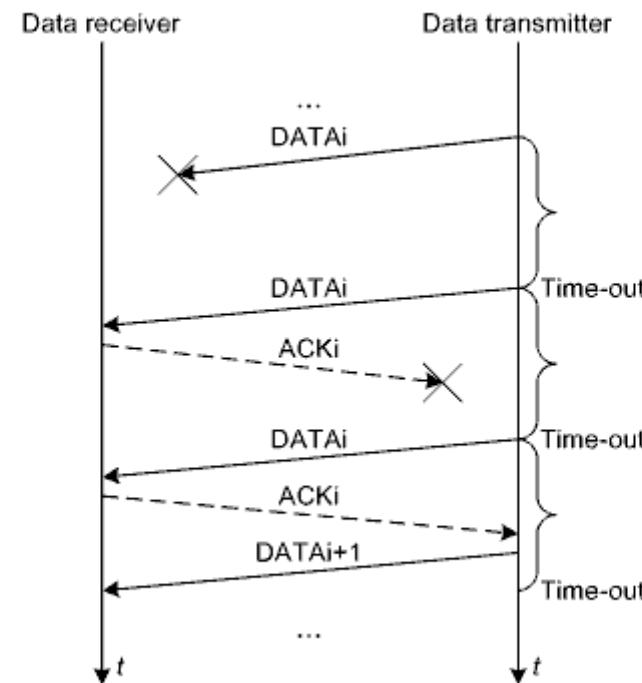
9.0.2.4

На практике, метод запросов-подтверждений невозможно реализовать без одного существенного дополнения.

Функционирование механизма запросов-подтверждений подразумевает ожидание определенных событий. Ожидание, в любом случае, не должно затягиваться до бесконечности.

Ограничение ожидания во времени достигается за счет применения *тайм-аута* (time-out). После передачи некоторого служебного или информационного пакета, требующего подтверждения, запускается таймер с обратным отсчетом. Если в течение заданного интервала времени соответствующая квитанция не приходит, то пакет считается утерянным и передается повторно (retransmission). Если квитанция не приходит снова и снова, то после некоторого конечного количества попыток дальнейшая передача считается бесперспективной и прекращается.

9.0.2.5



Пример взаимодействия с учетом тайм-аута

9.0.2.6

Следует учитывать, что:

- теряются могут как информационные пакеты, так и квитанции и пакеты-запросы;
- если квитанция приходит позже наступления тайм-аута, то этот факт приравнивается к ее потере;
- оптимальное время ожидания квитанций, применительно к некоторой СПД, зависит от ее особенностей.

9.0.3.1

В случае, когда СПД загружена незначительно, а взаимодействующие абоненты расположены далеко друг от друга, задействование классического механизма запросов-подтверждений приводит к неэффективному использованию ресурсов. Время, затрачиваемое на ожидание квитанций, становится недопустимо большим в сравнении с временем, затрачиваемым на передачу полезных данных.

Оптимизировать обмен позволяет применение оконного (window) метода, суть которого состоит в том, что до перехода к ожиданию квитанций передается не один, а несколько пакетов.

9.0.3.2

Выделяют два основных критерия классификации оконных методов.

Исходя из количества пакетов, передаваемых в окне, оно может быть:

1. *Статическим* (*static*) -- неизменяемый размер окна заложен в протокол или устанавливается на весь сеанс обмена.
2. *Динамическим* (*dynamic*) -- размер окна может изменяться (увеличиваться или уменьшаться) в процессе передачи сообщения.

Исходя из способа обработки очереди пакетов, окно может быть:

1. *Фиксированным* (*fixed*) -- перед формированием следующего окна текущее должно быть полностью «закрыто», то есть должны быть приняты все необходимые квитанции.
2. *Скользящим* (*sliding*) -- существует возможность сдвигать окно относительно последовательности пакетов.

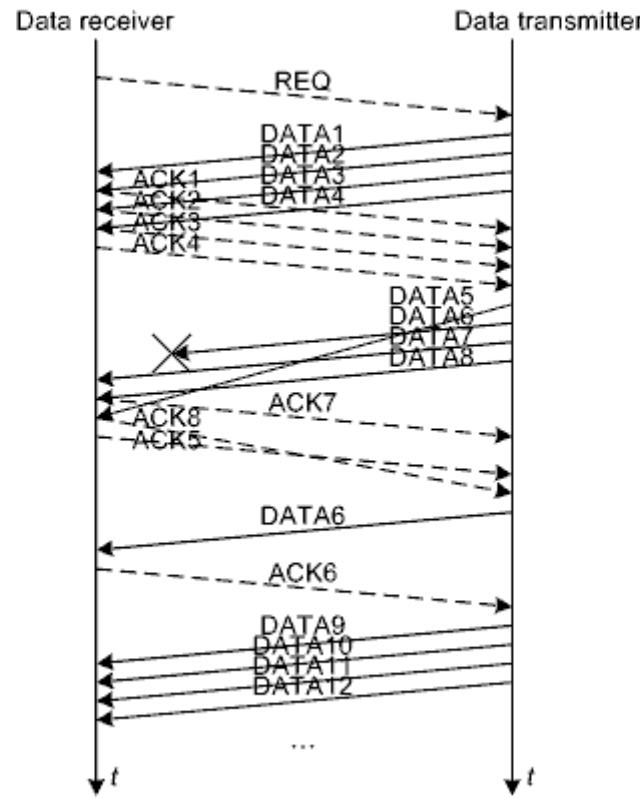
9.0.3.3

При реализации оконного метода следует учитывать следующие дополнительные обстоятельства:

- нужна нумерация пакетов в том или ином виде;
- подтверждаться может как все окно, так и каждый из пакетов;
- размером окна может управлять как передатчик, так и приемник;
- размером окна можно управлять посредством служебных полей, в том числе и в информационных пакетах;
- окно, с которым работает передатчик, может отличаться от окна, с которым работает приемник;
- иногда важен порядок доставки пакетов.

9.0.3.4

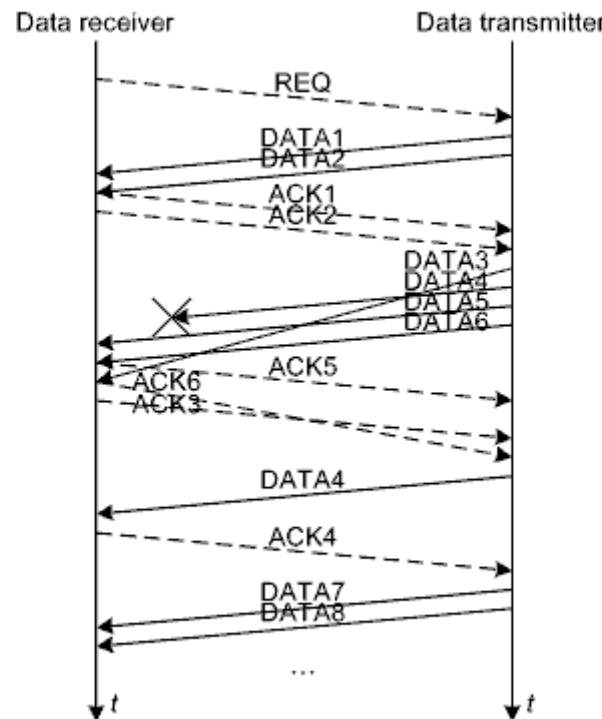
С точки зрения реализации, наиболее простым является статическое окно фиксированного размера.



Основной его недостаток состоит в отсутствии возможности адаптации к изменениям в СПД.

9.0.3.5

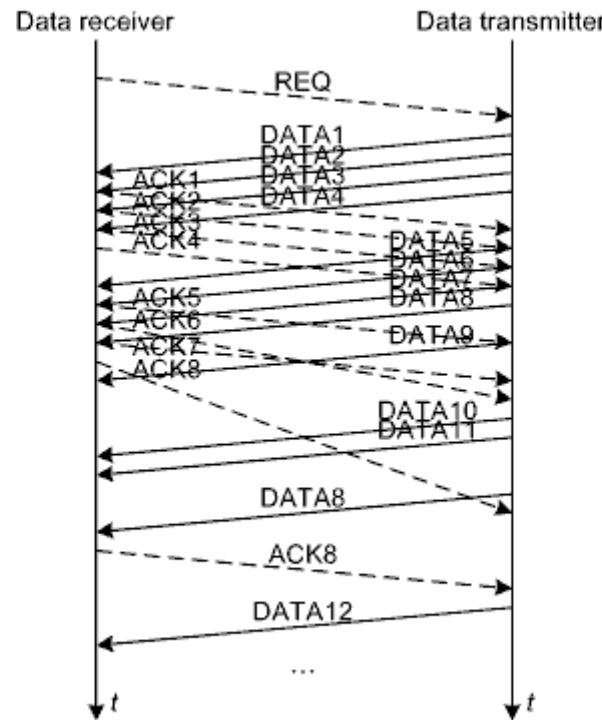
Первым вариантом усложнения является переход к динамическому окну.



Динамическое окно позволяет успешно адаптироваться к изменениям в СПД. При увеличении загруженности окно целесообразно сужать, а при снижении -- расширять.

9.0.3.6

Вторым вариантом усложнения является переход к скользящему окну.



Скользящее окно, особенно в сочетании с динамическим, позволяет ускорить адаптацию к топологическим и другим изменениям в СПД.

Таким образом, наиболее сложным является динамическое скользящее окно.

9.0.4.1

Классической реализацией оконного метода является оконный механизм протокола транспортного уровня TCP (Transmission Control Protocol) (основное RFC -- RFC 793).

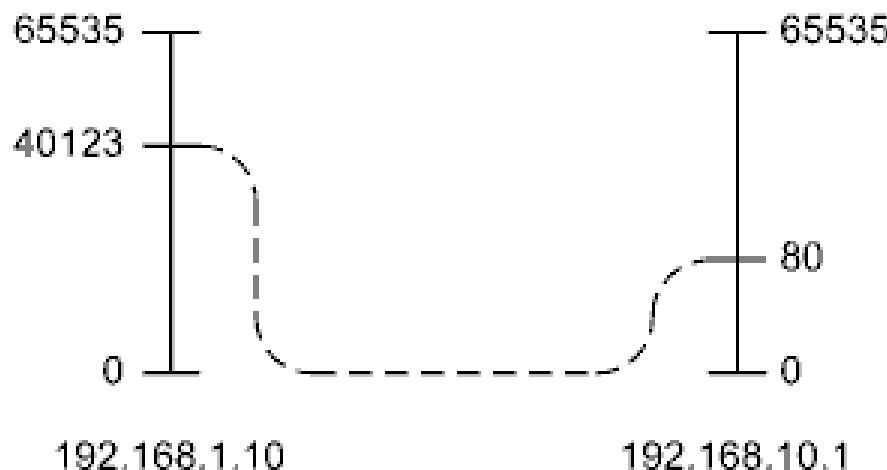
Протокол обеспечивает установление надежного соединения между сугубо пользовательскими или другими видами приложений, то есть доставка данных в правильном порядке гарантируется.

В стандарте TCP описано динамическое скользящее окно.

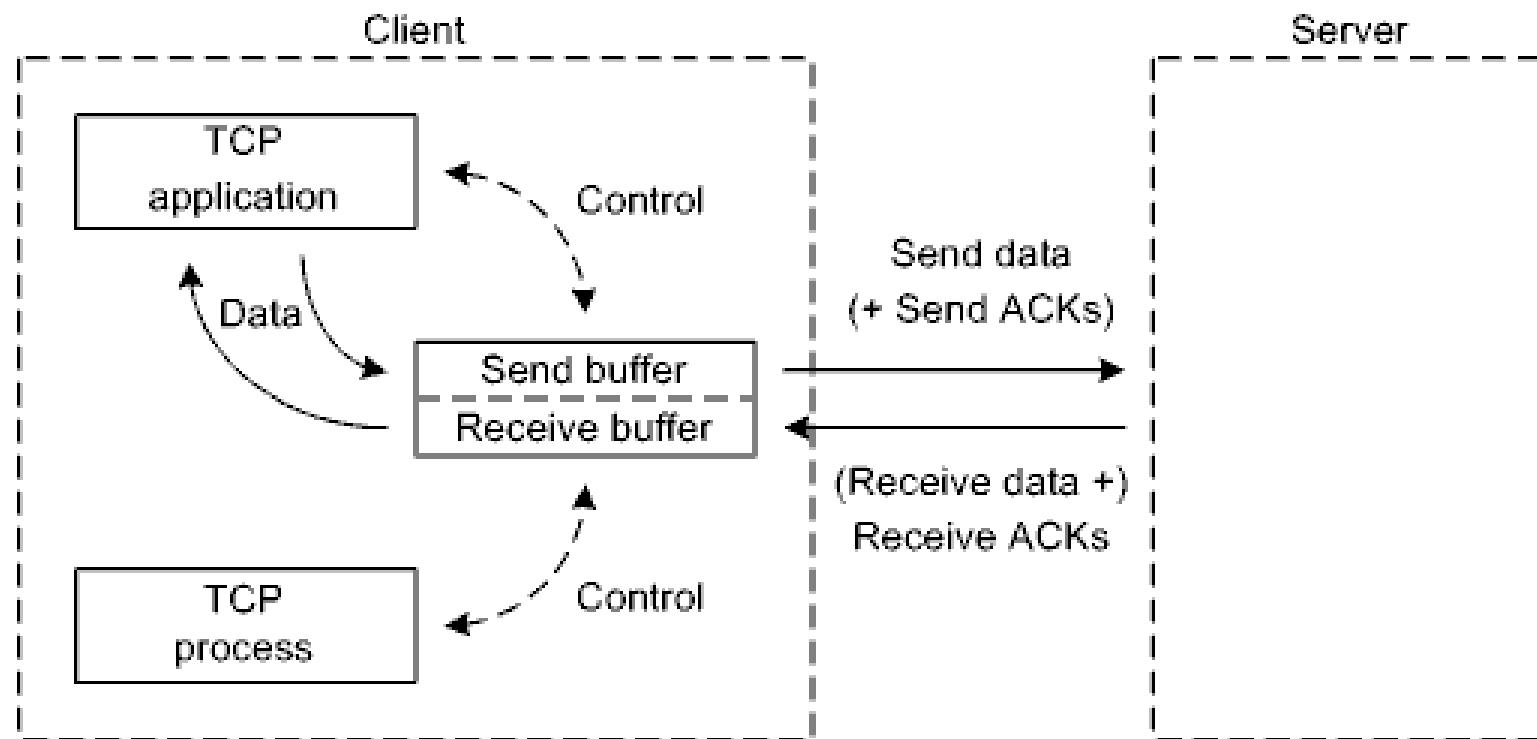
9.0.4.2

TCP соответствует клиент-серверной модели.

Сокет (socket) -- это «привязка» к виртуальному каналу, соединяющему между собой два взаимодействующих сетевых процесса, с точки зрения одного (любого) из этих процессов, причем с учетом всех трех уровней адресации.



9.0.4.3a



Структура соединения TCP

9.0.4.3b

Применительно к каждому TCP-соединению нужно выделять приложение, производящее или потребляющее сетевые данные, и TCP-процесс, предоставляющий коммуникационные услуги (например, специальный драйвер ОС).

Синхронизировать работу приложения и TCP-процесса можно только с помощью буферизации.

TCP-интерфейс, которым пользуется приложение, состоит из примитивов для работы с буфером, позволяющих контролируя записывать или считывать данные.

Доступ к буферу имеет и TCP-процесс, который отслеживает наполнение буфера и, используя ресурсы более низких уровней, организует прием или передачу данных.

9.0.4.4

Предназначенное для передачи сообщение разбивается на сегменты. Минимальной учитываемой в окне единицей данных является октет, то есть байт.

Все байты сообщения последовательно нумеруются так называемыми последовательными номерами -- SNs (Sequence Numbers).

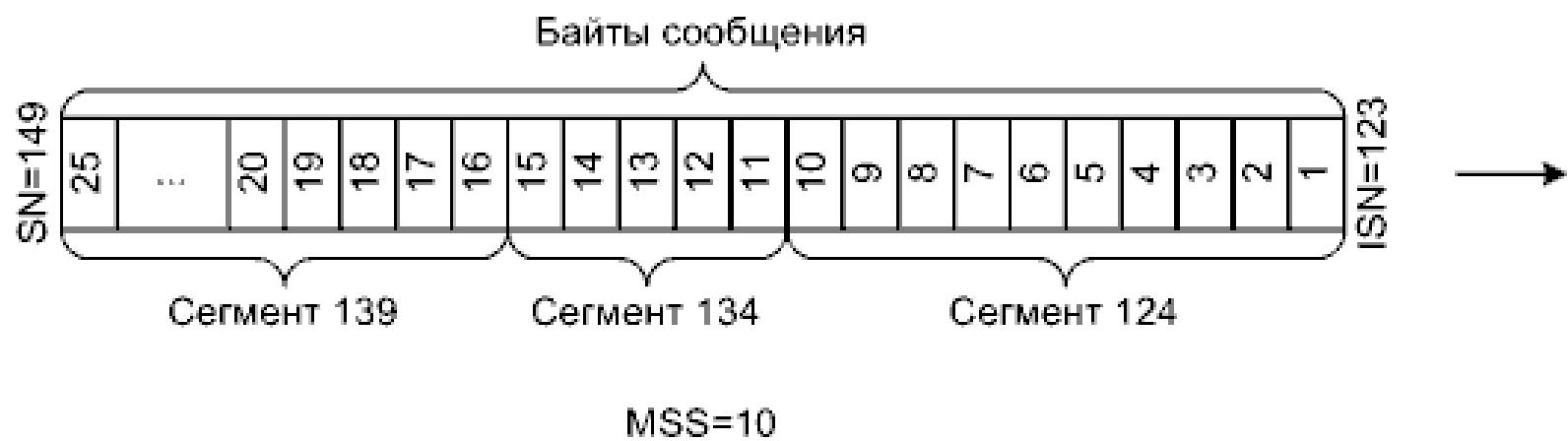
Нумерация начинается с некоторого начального последовательного номера -- ISN (Initial Sequence Number), который как правило не равен нулю, а генерируется реализациями случайно (например, на основе текущего времени) для того чтобы лучше управлять соединениями (например, после их ненормальных завершений).

Принято, что сам ISN в нумерацию байтов не включается, то есть номер первого байта сообщения больше ISN на единицу.

Номером сегмента является SN первого байта данных в нем.

По разным понятным причинам длина сегмента может варьировать, но она имеет ограничение. Поэтому важное значение имеет конфигурационный параметр MSS (Maximum Segment Size) -- максимальная длина сегмента (по умолчанию 536 байтов).

9.0.4.5



Пример сегментации сообщения

9.0.4.6

В стандарте выделяют несколько видов окон, которые нужно различать. Благодаря гибкости протокола, передающий и принимающий TCP-процессы работают с разными окнами, то есть, в первую очередь, следует отдельно рассматривать окно передачи (send window) и окно приема (receive window).

9.0.4.7а



Организация буфера передачи

9.0.4.7b

Передающее приложение последовательно, «порциями», записывает блоки байтов сообщения, возможно разной длины, в буфер передачи.

Длина сообщения и размер буфера -- это вещи независимые, они почти всегда различаются.

TCP-процесс формирует из имеющихся в буфере данных соответствующее количество сегментов и последовательно отправляет их.

В любой момент времени текущее окно (current window) передачи имеет некоторый установленный размер и характеризуется тем, что все попадающие в него сегменты с данными можно передавать без ожидания подтверждений.

Его правая (на рисунке) граница совпадает с правой границей буфера и скользит налево относительно последовательности сегментов с данными по мере поступления и упорядочивания подтверждений.

Переданные, но неподтвержденные сегменты с данными продолжают оставаться в буфере, так как возможно потребуется их повторная передача.

9.0.4.7c

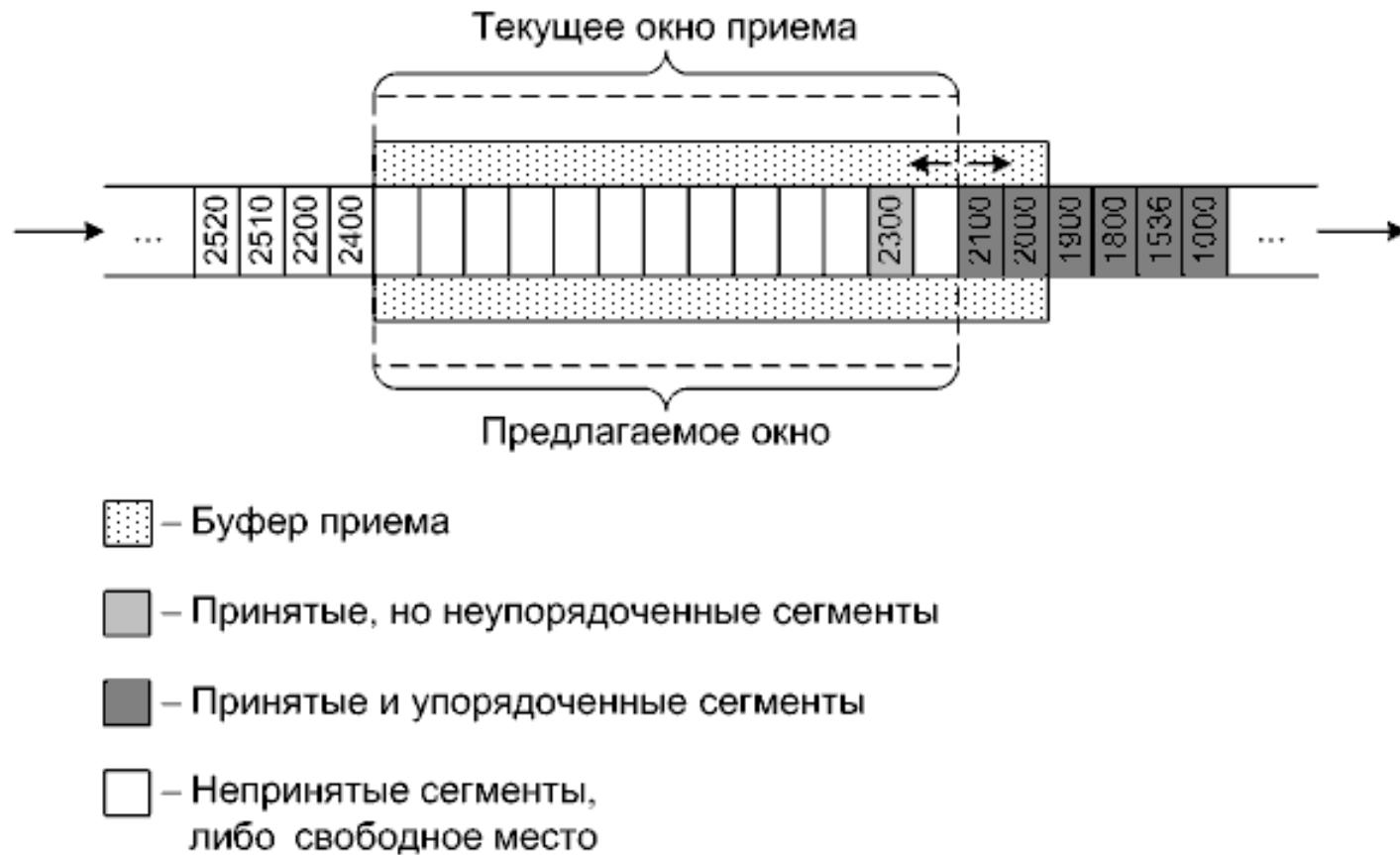
Левая граница «привязана» к правой в соответствии с размером текущего окна. Но поскольку размер подвержен динамической коррекции, положение левой границы относительно правой постоянно изменяется.

Область текущего окна передачи за вычетом переданных, но неподтвержденных сегментов с данными, является доступным окном (useable равно effective window).

TCP-процесс должен последовательно отправить все сегменты с данными, попавшие в эту область.

Если размер текущего окна передачи равен нулю, то передача приостанавливается полностью.

9.0.4.8а



Организация буфера приема

9.0.4.8b

На другой стороне соединения, возможно уже разупорядоченные при преодолении СПД сегменты поступают в буфер приема (размер может не совпадать с размером буфера передачи). При этом они размещаются там согласно своим номерам.

Текущее окно приема охватывает часть буфера, в которой можно размещать еще неупорядоченные сегменты с данными.

Как и текущее окно передачи, в любой момент времени оно так же имеет некоторый определенный размер.

Левая (на рисунке) граница текущего окна приема совпадает с левой границей буфера.

Правая граница проходит слева за последним упорядоченным сегментом с данными и поэтому динамически меняет свое положение относительно левой границы.

По мере считывания принимающим приложением упорядоченных байтов из буфера окно скользит относительно последовательности сегментов с данными.

Если размер текущего окна приема равен нулю, а сегменты с данными продолжают поступать, то возникает переполнение.

9.0.4.8с

Вполне закономерно, что именно на принимающий TCP-процесс, как на более подверженный влиянию недетерминированности СПД, возложен контроль «поведения» оконного механизма. Это делается посредством «обратной связи». Принимающий TCP-процесс пытается информировать передающий о состоянии своего буфера, точнее о наличии в нем свободного места. Для этого он при подтверждениях сообщает предлагаемое окно (*announced* равно *advertised* равно *offered window*).

В качестве размера предлагаемого окна указывается размер текущего окна приема. Последствия разупорядочивания сегментов с данными такому подходу не противоречат.

9.0.4.9

Максимальный размер любого из окон не может превышать размер соответствующего буфера (например, 8 килобайтов).

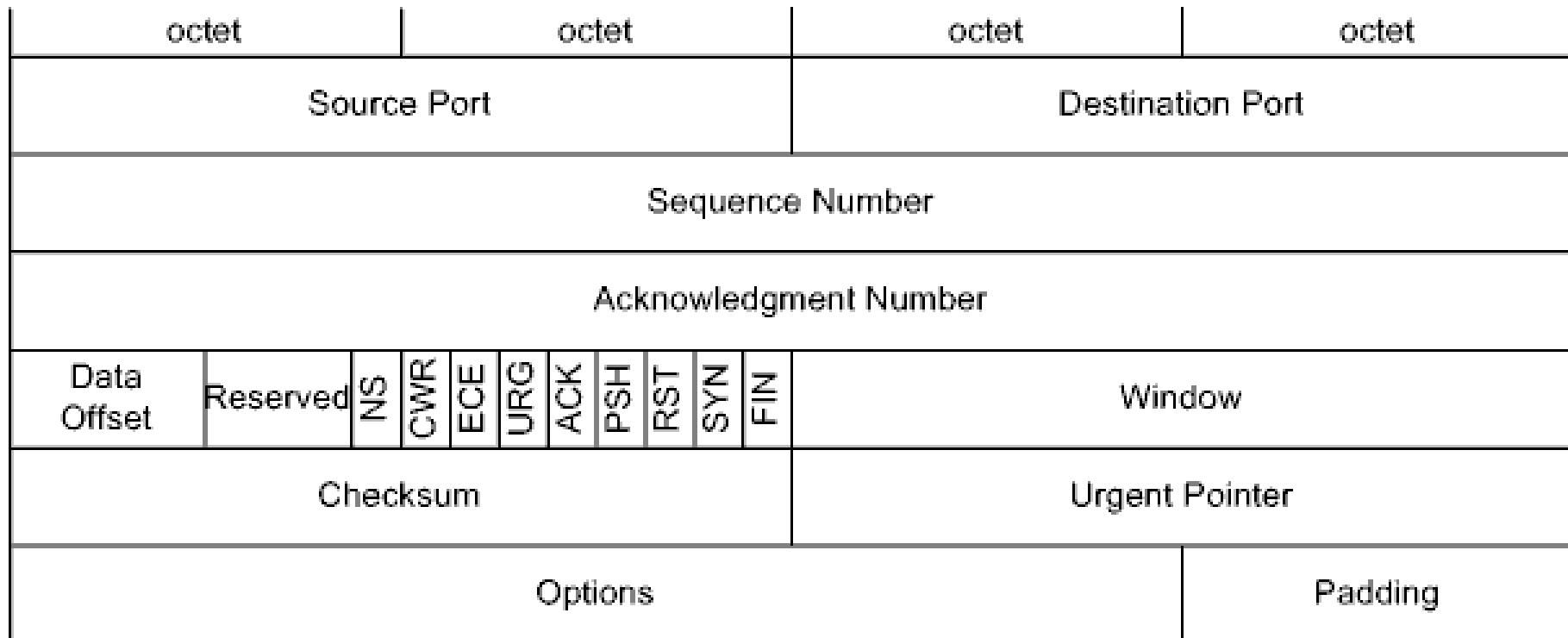
9.0.4.10

В результате, можно сделать вывод о том, что на работу соединения влияют приложения, TCP-процессы и сетевой уровень.

В идеале, при полностью сбалансированной работе, размер текущего окна передатчика равен размеру предлагаемого окна, то есть равен размеру текущего окна приемника. А если еще и буферы освобождаются «мгновенно», то этот размер совпадает с размером доступного окна и размерами буферов.

Алгоритмы TCP направлены на «уравнивание» всех упомянутых окон.

9.0.4.11a



Формат заголовка TCP

9.0.4.11b

Поля:

1. Source Port -- программный порт источника.
2. Destination Port -- программный порт назначения.
3. Sequence Number (SN) -- последовательный номер (сегмента).
4. Acknowledgment Number (AN) -- подтверждающий номер.
5. Data Offset -- смещение данных (в 32-ухбитных словах).
6. Reserved -- зарезервировано (должно равняться нулю).
7. URG (URGent Pointer field significant) -- флаг значимости указателя на экстренные данные.
8. ACK (ACKnowledgment field significant) -- флаг значимости подтверждающего номера.
9. NS (Nonce Sum) -- флаг -- контрольная сумма для проверки правильности кодов явных уведомлений о заторах (связан с QoS, связан с IP-заголовком) (RFC 3540).
10. CWR (Congestion Window Reduced) -- флаг уменьшения окна затора при явном уведомлении о заторе (RFC 3168).
11. ECE (Explicit Congestion Notification Echo) -- флаг подтверждения явного уведомления о заторе (RFC 3168).

9.0.4.11c

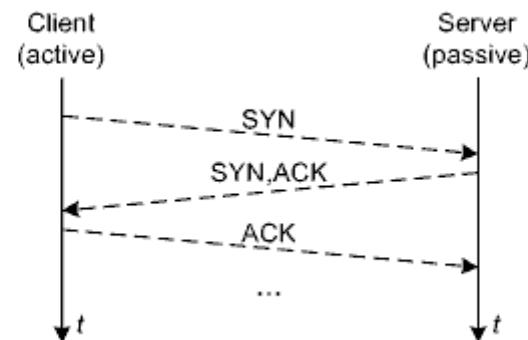
12. PSH (PuSH Function) -- флаг принудительной доставки данных (без буферизации).
13. RST (ReSeT the connection) -- флаг разрыва соединения (например, из-за сбоя на одной из взаимодействующих сторон).
14. SYN (SYNchronize sequence numbers) -- флаг синхронизации последовательных номеров.
15. FIN (No more data from sender) -- флаг последних данных.
16. Window (W) -- предлагаемое окно.
17. Checksum -- контрольная сумма.
18. Urgent Pointer -- указатель на экстренные данные (RFC 6093).
19. Options -- опции (например, MSS).
20. Padding -- наполнитель.

9.0.4.12

Функционирование оконного механизма TCP базируется на использовании трех полей в заголовке сегмента: SN, AN, W, и трех флагов (из шести стандартизованных изначально): SYN, ACK, FIN.

9.0.4.13а

Установление TCP-соединения, известное как «тройное рукопожатие» (three-way handshake), основывается на использовании флагов SYN и ACK.



(На этом и последующих рисунках указаны ключевые задействованные флаги и поля. Сплошной линией обозначены сегменты с данными, пунктирной -- сугубо служебные.)

9.0.4.13b

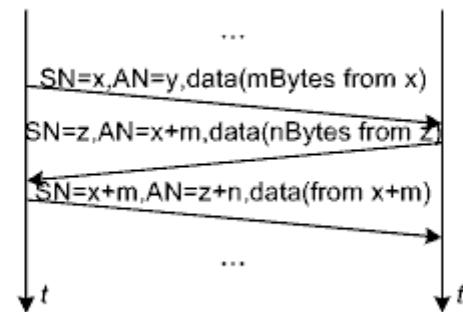
Сначала TCP-процесс -- инициатор взаимодействия (на стороне клиента) отправляет служебный сегмент с установленным флагом SYN, тем самым сообщая о своих намерениях (первое «рукопожатие»).

Затем запрашиваемый TCP-процесс (на стороне сервера), если он согласен взаимодействовать, подтверждает это ответным служебным сегментом с двумя установленными флагами SYN и ACK (второе «рукопожатие»).

Наконец, инициатор отвечает еще одним служебным сегментом с установленным флагом ACK, тем самым подтверждая подтверждение (третье «рукопожатие»).

9.0.4.14

Не смотря на то, что процесс установления соединения несимметричен, в дальнейшем, в общем случае, оно используется в полнодуплексном режиме.



9.0.4.15

Очень важно, что на обмен сегментами нужно смотреть с двух сторон.

При этом один и тот же TCP-процесс, находящийся по одну сторону соединения, одновременно может выступать в качестве как передатчика данных, так и приемника данных.

Полнодуплексность самого соединения достигается за счет того, что передаваемый в определенном направлении сегмент служит одновременно для транспортировки как данных и связанных с ними служебных полей от передающей составляющей TCP-процесса, так и подтверждений и связанных с ними других служебных полей от принимающей составляющей TCP-процесса.

В СПД одновременно могут находиться множество сегментов, относящихся к одному соединению.

Применительно к данным в одном сегменте, соединение является полудуплексным, так как сегмент не может содержать более одного поля с ними.

9.0.4.16

По правилу протокола, поле SN пересылаемого сегмента отражает собственный SN этого сегмента.

По другому правилу, в поле AN указывается SN ожидаемого сегмента, коим является следующий по порядку сегмент.

При установлении соединения данные не пересылаются. Поэтому, для того чтобы не нарушать указанные правила, в качестве SNs используют невключенные в нумерацию байтов сообщения ISNs, а в качестве ANs -- просто инкрементированные SNs. Обойтись без передачи SNs при установлении соединения невозможно, так как стороны должны однозначно идентифицировать это соединение.

После синхронизации SNs соединение считается установленным (established).

9.0.4.17

Флаг SYN используется только при установлении соединения, а флаг ACK -- в каждом ответном сегменте.

9.0.4.18

Не смотря на предоставляемые возможности, данные вполне могут пересылаться только в одном направлении, то есть в симплексном режиме.

При этом в направлении, попутном направлению пересылки данных, в качестве AN используется SN следующего по порядку несуществующего (вообще, либо уже, либо пока) сегмента, что никоим образом не противоречит уже приведенным правилам.

Если сегментов с данными пересылается несколько, то ANs дублируются столько раз, сколько нужно.

Это приводит к дублированию SNs в ответных сегментах без данных.

Аналогичные дублирования возникают и при приостановке пересылки данных в определенном направлении.

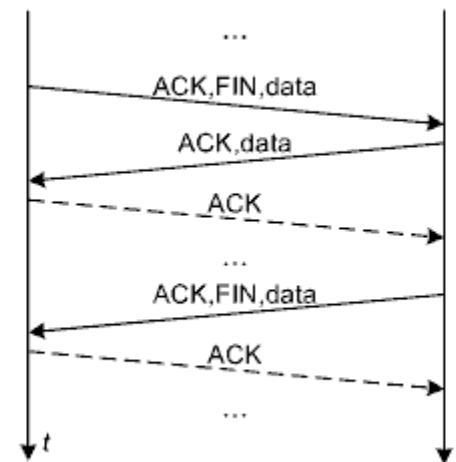
9.0.4.19

Почему нельзя инкрементировать SNs при посылке сегмента без данных?

9.0.4.20

Поскольку при установлении соединения оно всегда открывается в двух направлениях (по инициативе клиента, но может использоваться в одном любом направлении), для нормального завершения оно и закрыто должно быть в обоих направлениях.

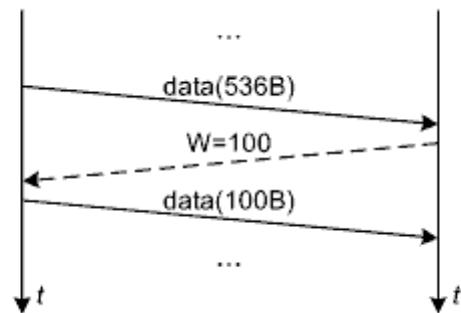
Для закрытия соединения в своем направлении, сторона, в соответствующем сегменте (обычно с последними данными), устанавливает флаг FIN.



Соединение, нормально закрытое только в одном направлении, или ненормально завершенное на одной из сторон без уведомления другой стороны (в результате сбоя) называют полуоткрытым (half-open).

9.0.4.21

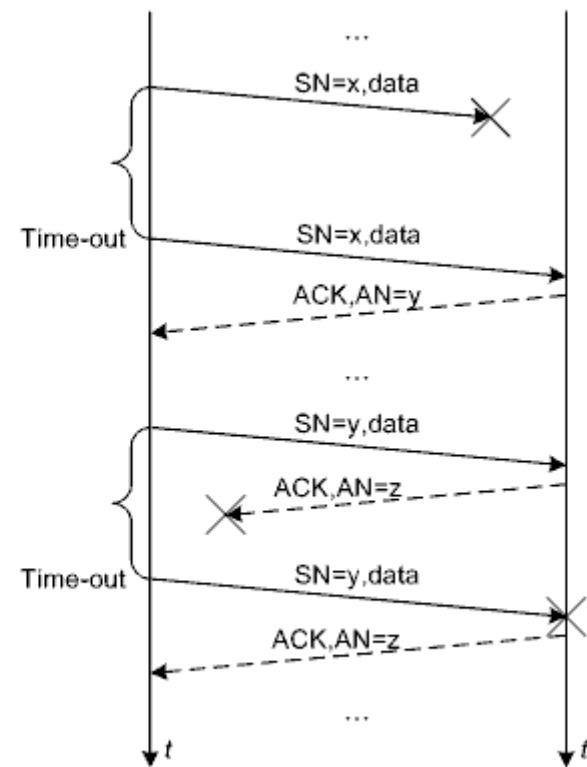
Размер предлагаемого окна в поле W может изменяться каждый раз для соответствующей коррекции текущего окна передачи, в том числе и при установлении соединения для изменения размера текущего окна передачи по умолчанию.



В случае задания нулевого значения поля W передача данных фактически запрещается. После освобождения места в буфере приема подтверждение обязательно повторяется с уже ненулевым полем W, что «разблокирует» передающую сторону.

9.0.4.22а

Проблема возможной потери в СПД некоторых сегментов решается с помощью тайм-аутов.



9.0.4.22b

Передающий TCP-процесс определяет потерю сегмента с данными либо его подтверждения по отсутствию этого подтверждения в течение установленного интервала времени. После наступления тайм-аута сегмент с данными передается повторно.

Отрицательные подтверждения не предусмотрены вообще. Принимающий TCP-процесс подтверждает все принятые сегменты с данными, причем подтверждает всегда. При этом если принята копия (что говорит о потере подтверждения), то она удаляется.

Получение сегмента с SN больше ожидаемого говорит о возможной потере сегментов с данными или о разупорядочивании.

9.0.4.23

Важно правильно оценивать время «отклика системы». Поэтому время ожидания подтверждений рассчитывается на основе показаний таймеров и корректируется.

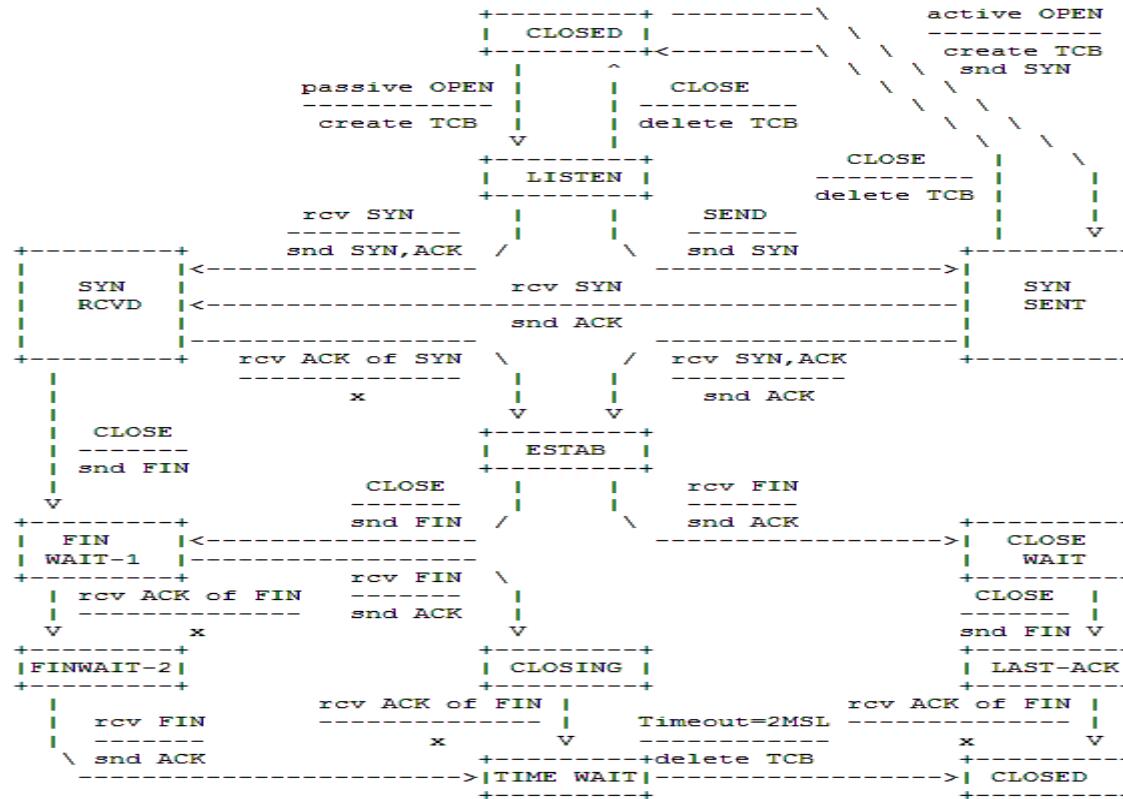
При этом первостепенное значение имеет параметр RTT (Round-Trip Time) -- суммарное время пересылки по СПД сегмента с данными и его подтверждения.

9.0.4.24

Протокол TCP обладает несколькими дополнительными возможностями. Возможна пересылка экстренных данных (urgent data) и ускоренная пересылка (push function).

9.0.4.25

Состояния TCP-процесса стандартизированы и предусмотрена диаграмма переходов между состояниями.



Для обеспечения своей функциональности TCP-процесс должен хранить множество ассоциированных с каждым соединением служебных данных.

[RFC]

9.0.4.26

Как было сказано выше, несбалансированность соединений может возникать из-за разной производительности задействованных подсистем, плюс из-за изменения производительности этих подсистем.

Следует отметить, что базовая редакция стандарта TCP предоставила реализациям определенную вольность (как во многих других случаях со стандартами). Это привело к тому, что при полном соблюдении требований во многих случаях оконный механизм TCP может оказаться неэффективным. «Разношерстность» реализаций усугубляет ситуацию.

Как следствие, потребовалась разработка дополнений к базовому алгоритму. Новые алгоритмы оперируют с новыми понятиями.

9.0.4.27

В частности, хорошо известна проблема, вошедшая в историю под обобщенным названием «синдром глупого окна» («silly window syndrome»), в свое время «стопорившая» значительную часть пространства Internet. Синдром может возникать по разным причинам и проявляется в том, что текущее окно передачи не соответствует состоянию приемника, тем самым не позволяя его как следует «нагрузить» либо, наоборот, «разгрузить».

Решение Нэгла (Nagle) позволяет побороть «синдром глупого окна» когда передающей стороне требуется часто отправлять небольшие сегменты с данными.

Решение Кларка (Clark) позволяет побороть «синдром глупого окна» когда принимающей стороной часто анонсируется небольшое предлагаемое окно.

9.0.4.28а

Также стандартизированы четыре дополнения Ван Якобсона (Van Jacobson), призванные бороться с перегрузками в СПД (последнее RFC -- RFC 5681):

1. Медленный старт (slow start).

Идея заключается в том, что в начале передачи размер текущего окна передачи нужно увеличивать не «скачком», а плавно, пропорционально скорости получения подтверждений (не превышая размер предлагаемого окна).

9.0.4.28b

Рекомендуемые формулы:

$IW = 2 * SMSS$, если $SMSS > 2190$ Bytes ,

$IW = 3 * SMSS$, если 2190 Bytes $\geq SMSS > 1095$ Bytes ,

$IW = 4 * SMSS$, если $SMSS \leq 1095$ Bytes ,

где IW (initial window) -- начальное значение текущего окна передачи:

$cwnd += \min(N, SMSS)$,

где $cwnd$ (congestion window) -- текущее окно передачи (в данном случае, окно затора), N -- количество подтвержденных байтов, $SMSS$ (sender MSS) -- MSS передатчика.

9.0.4.29

2. Избегание затора (congestion avoidance).

Состоит в сдерживании экспоненциального роста размера текущего окна передачи после преодоления им некоторого порога. Как правило переход к избеганию затора происходит после медленного старта.

Рекомендуемые формулы:

$$ssthresh = \max (FlightSize / 2, 2 * SMSS) ,$$

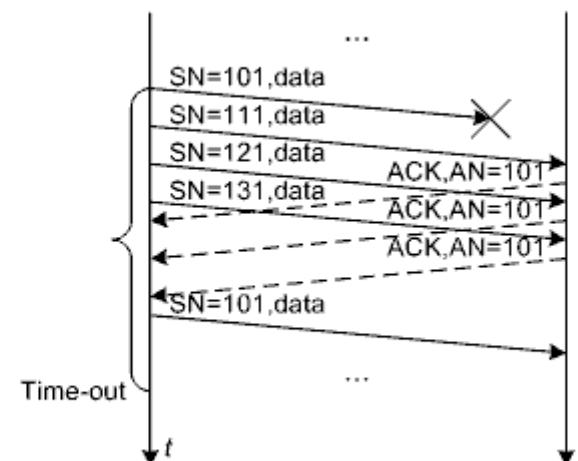
где $ssthresh$ (slow start threshold) -- порог перехода от медленного старта к избеганию затора, $FlightSize$ -- количество еще неподтвержденных байтов;

$$cwnd += SMSS * SMSS / cwnd .$$

9.0.4.30

3. Быстрая повторная передача (fast retransmit).

При получении принимающей стороной разупорядоченного сегмента с данными (возможно из-за потери ожидаемого сегмента с данными) незамедлительный повтор подтверждения с AN недостающего сегмента с данными. При получении передающей стороной трех одинаковых подтверждений незамедлительный повтор сегмента с данными согласно AN. Что, в некоторых ситуациях, позволяет успешно переслать потерянный сегмент еще до наступления тайм-аута.

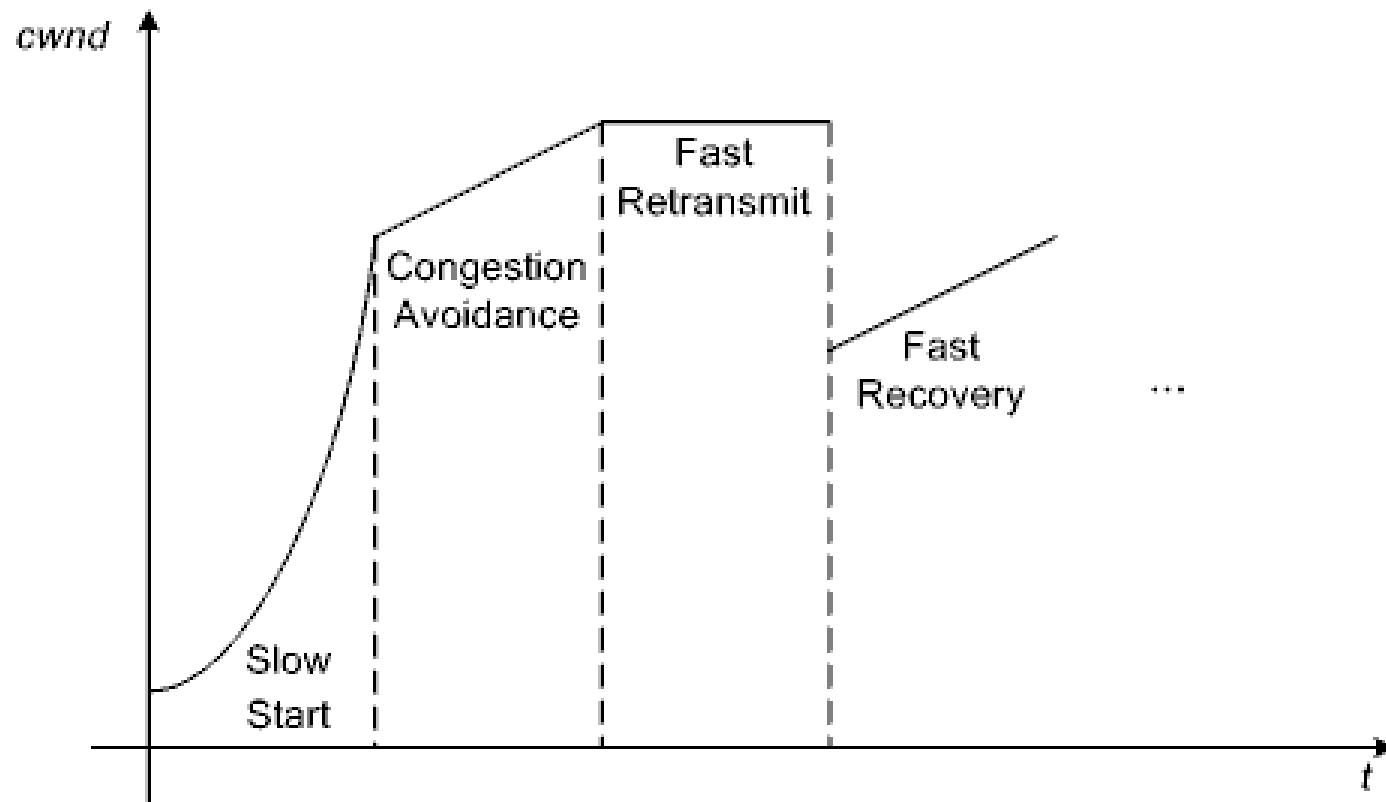


9.0.4.31

4. Быстрое восстановление (fast recovery).

После обнаружения затора, переход сразу к избеганию коллизий, минуя стадию медленного старта. Как правило в связке с быстрой повторной передачей.

9.0.4.32



TCP **congestion control**

9.0.4.33

Последствия потерь и разупорядочивания сегментов заключаются в разрушении «маятника» взаимодействия и приводят к необходимости еще одной важной оптимизации, четко проявляющейся при быстрой повторной передаче.

Согласно базовому алгоритму все сегменты должны быть подтверждены, а значит, после быстрой повторной передачи принимающая сторона должна послать все недостающие подтверждения.

Но стороны могут «договориться», что текущий AN отражает номер первого ожидаемого получателем сегмента плюс автоматически подтверждает все сегменты с меньшими номерами (*cumulative acknowledgement*).

9.0.4.34

Были разработаны и другие усовершенствования, основанные, например, на манипулировании с RTT (RFC 6298, RFC 7323).

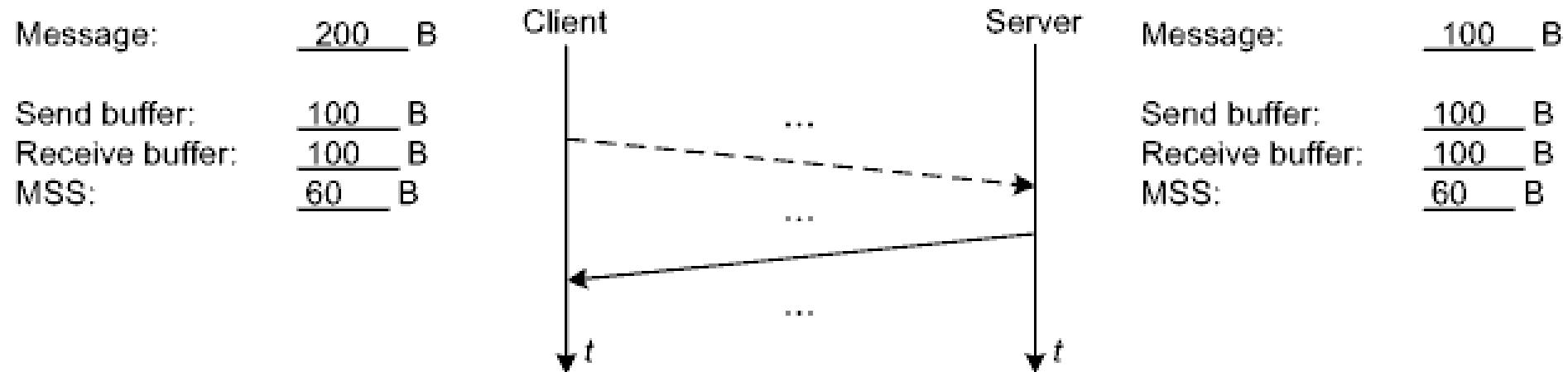
9.0.4.35

Задача у доски.

Необходимо переслать два сообщения указанного размера, одно в направлении от клиента к серверу, другое в противоположном направлении.

Указаны также требующиеся конфигурационные параметры TCP.

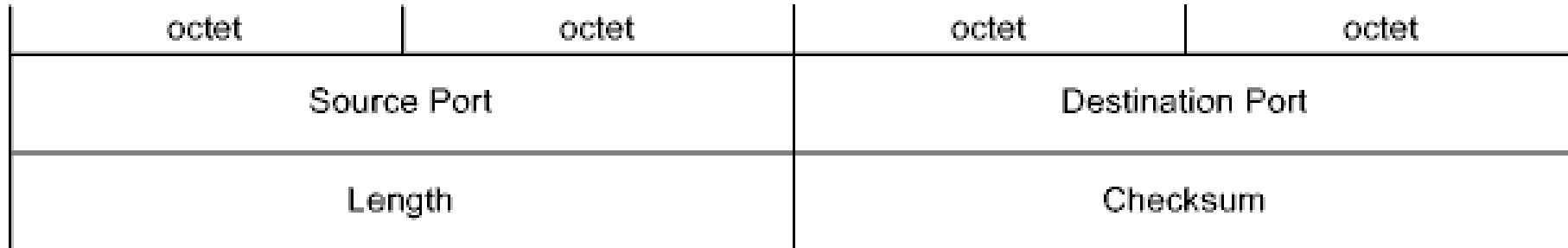
Нарисуйте диаграмму взаимодействия между клиентом и сервером с указанием значений полей SN, AN, W в TCP-заголовке, флагов SYN, ACK, FIN в TCP-заголовке и количества данных в сегментах.



9.0.5.1

Протокол транспортного уровня UDP (User Datagram Protocol) (RFC 768) реализует способ пересылки данных без гарантии доставки, часто называемый *дейтаграммным* (datagram) (хотя user datagram -- это пакет с контролируемыми пользователем данными, а datagram -- это любой пакет с данными).

9.0.5.2



Поля:

Source Port -- программный порт источника.

Destination Port -- программный порт назначения.

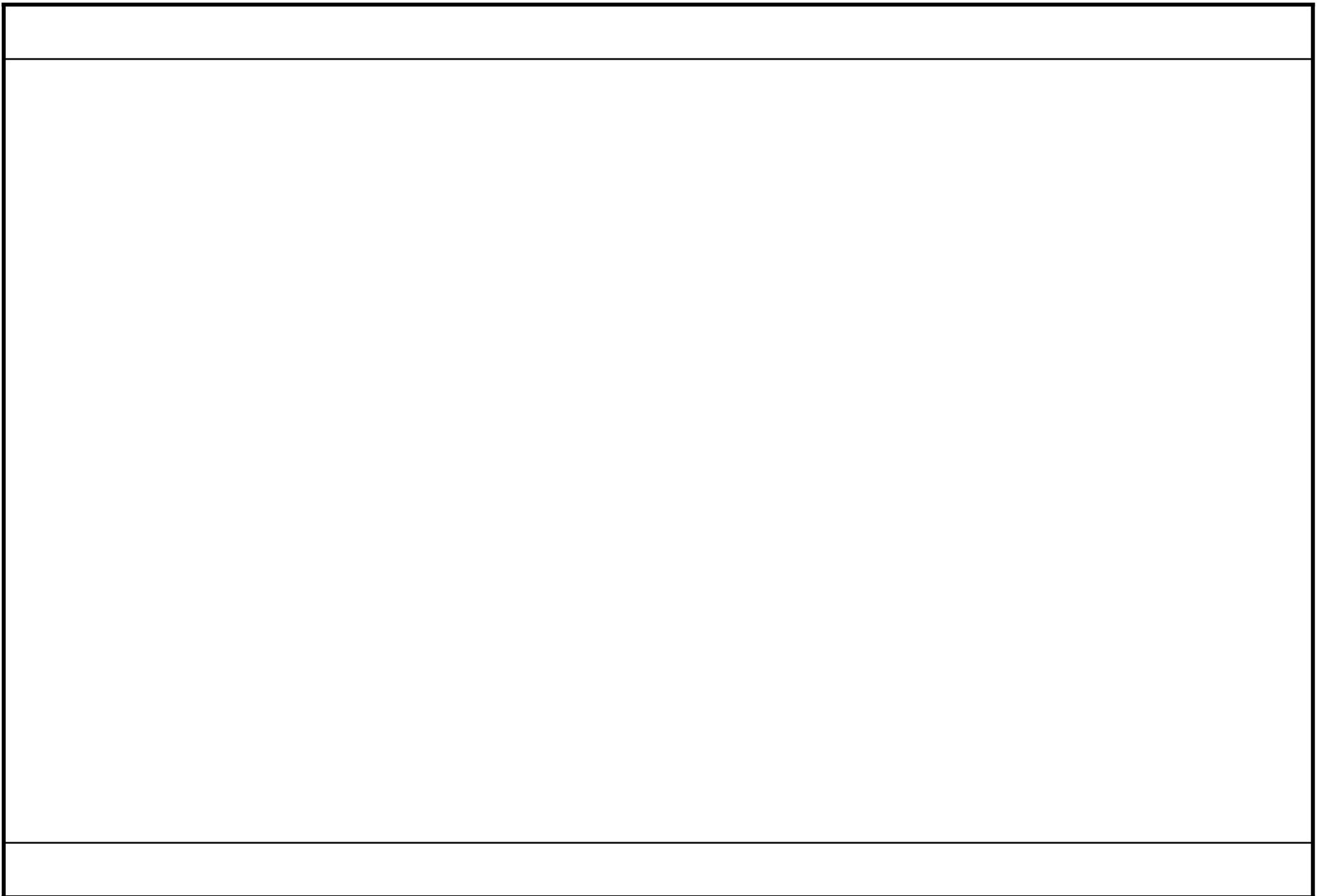
Length -- длина дейтаграммы включая заголовок (в байтах).

Checksum -- контрольная сумма (псевдозаголовка, плюс заголовка, плюс данных).

Формат заголовка UDP

9.0.5.3

При вкладывании UDP-дейтаграммы в IP-пакет (IPv4, IPv6), между UDP-заголовком и IP-заголовком вставляется дополнительный так называемый UDP-псевдозаголовок, в котором дублируются некоторые значения из основного IP-заголовка.



ПРИКЛАДНЫЕ ЗАДАЧИ В КОМПЬЮТЕРНЫХ СЕТЯХ

10.0.1.1

В рамках данной дисциплины предполагается рассмотрение следующих классических прикладных протоколов семейства TCP/IP:

1. FTP.
2. Telnet.
3. SMTP, POP, IMAP.
4. HTTP.

Для подробного рассмотрения безопасности прикладных протоколов предусмотрена отдельная дисциплина, поэтому соответствующий материал во многом опущен.

(В таблицах ниже серым цветом выделена информация для поверхностного изучения. Аргументы команд так же приведены для ознакомления -- без подробного описания.)

10.1

FTP

Версия 2.4

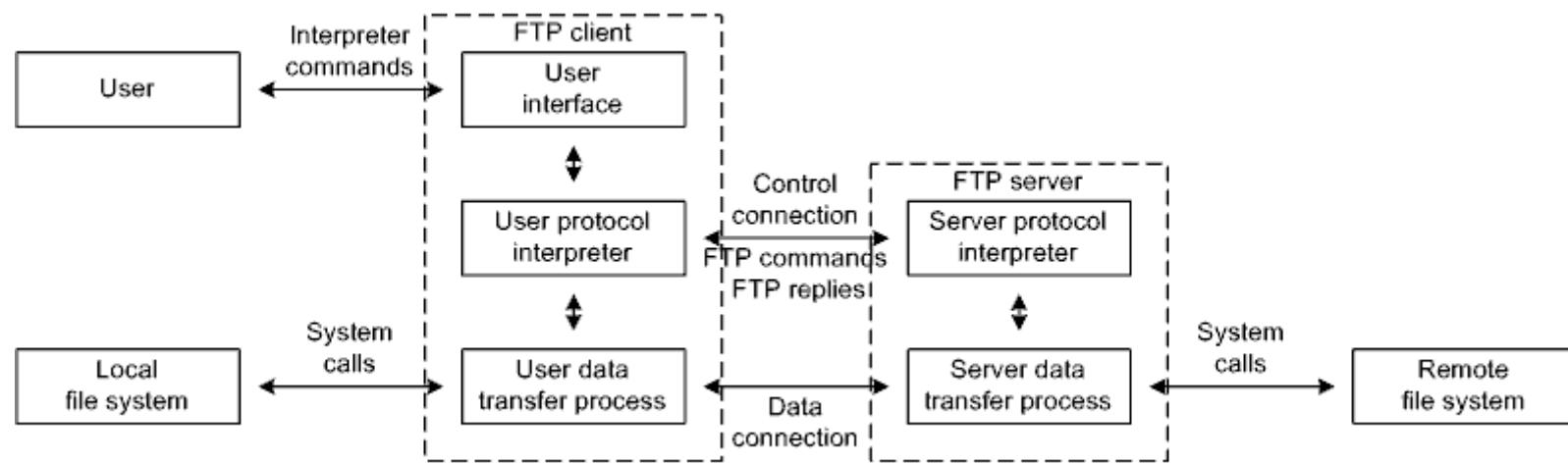
10.1.1.1

Как и следует из его названия, протокол FTP (File Transfer Protocol) (основное RFC -- RFC 959) предназначен для пересылки файлов между двумя удаленными станциями.

FTP разрабатывался одним из первых, но до сих пор занимает значимое место в сети Internet.

FTP базируется на клиент-серверной модели и использует транспорт TCP.

10.1.1.2



Структура системы FTP

10.1.1.3

FTP-клиент обслуживает запросы пользователя и работает на локальной по отношению к нему станции.

FTP-сервер обслуживает запросы FTP-клиента и работает на удаленной станции.

На рисунке показана взаимосвязь между одним FTP-клиентом и одним FTP-сервером, но возможна также схема взаимодействия когда по инициативе FTP-клиента осуществляется файловый обмен между двумя FTP-серверами.

Как в составе FTP-клиента, так и в составе FTP-сервера **выделяют** соответствующие протокольные интерпретаторы (protocol interpreters) и процессы пересылки данных (data transfer processes).

10.1.1.4

FTP **относят** к протоколам, ориентированным на пользователя. Это означает, что реализация, по крайней мере FTP-клиента, обязана предоставлять пользователю более или менее функционально полный интерфейс.

Классический интерфейс FTP-клиента, широко применяющийся в оболочках UNIX и соответствующих окнах Windows, представляет собой интерпретатор командной строки, активизируемый вводом команды `ftp`. В качестве аргументов можно задать название либо IP-адрес FTP-сервера, а также номер порта, если он отличен от стандартного. Если команда введена успешно, появится приглашение интерпретатора:

```
ftp>
```

Существуют также множество прикладных программ и пакетов, использующих графический интерфейс.

10.1.1.5

FTP-сервер представляет собой непрерывно выполняющуюся программу, ожидающую запросы от FTP-клиентов, выраженную в виде демона UNIX либо сервиса Windows.

В ОС UNIX работа демонов обычно контролируется конфигурационными файлами, а в Windows -- соответствующими оконными средствами.

10.1.1.6

В отличие от многих других протоколов, FTP задействует не одно, а два соединения, значит для него зарезервированы два номера программных портов (на стороне FTP-сервера):

20 -- FTP Data -- информационное соединение (data connection).

21 -- FTP -- управляющее соединение (control connection).

Сначала FTP-клиентом создается управляющее соединение, которое в дальнейшем используется только для передачи FTP-команд от FTP-клиента и FTP-ответов от FTP-сервера. FTP-сервер принимает, интерпретирует и выполняет FTP-команды, а также передает FTP-ответы.

Одно или несколько информационных соединений, предназначенных исключительно для пересылки данных, то есть файлов и каталогов, создаются FTP-сервером или FTP-клиентом. Они не существуют на протяжении всего сеанса взаимодействия и могут создаваться и ликвидироваться по мере необходимости. Управляющее же соединение может быть завершено только после осуществления полезного информационного обмена, если таковой нужен.

В некоторых особых ситуациях может происходить отказ от использования стандартных портов.

10.1.1.7

Можно выделить три уровня, связанных с применением FTP:

1. Настройка, запуск и использование пользователем FTP-клиента, а администратором -- FTP-сервера.
2. Работа пользователя с протокольным интерпретатором.
3. Скрытое от пользователя взаимодействие непосредственно по протоколу FTP.

На каждом из этих уровней существует свое понятие «команда».

На самом высоком уровне это команда ОС.

На промежуточном уровне это уже команда, вводимая при работе с программой FTP-клиента, то есть команда интерпретатора.

И, наконец, на низком уровне это собственно команда протокола, передаваемая через управляющее соединение, то есть FTP-команда.

Некоторые аббревиатуры команд интерпретатора и FTP-команд совпадают. Но необходимо понимать, что аббревиатуры все-таки отличаются, и учитывать тот факт, что одна команда интерпретатора может реализовываться последовательностью из нескольких FTP-команд.

10.1.2.1

FTP-команда представляет собой последовательность из трех-четырех букв, за которыми могут следовать аргументы.

Регистр букв не учитывается.

Аргументы отделяются пробелами (<SP>).

FTP-команда завершается парой символов возврата каретки и перевода строки (<CRLF>).

(В квадратные скобки заключены опциональные аргументы.)

10.1.2.2а

FTP-команда	Название	Описание
Access control commands (команды контроля доступа)		
USER <SP> <username> <CRLF>	USER NAME	Имя пользователя
PASS <SP> <password> <CRLF>	PASSWORD	Пароль (должна следовать непосредственно за USER)
ACCT <SP> <account- information> <CRLF>	ACCOUNT	Пользовательский аккаунт (альтернатива паре USER и PASS при доступе к специфическим ресурсам)
CWD <SP> <pathname> <CRLF>	CHANGE WORKING DIRECTORY	Сменить рабочий каталог (текущий каталог удаленной файловой системы)
CDUP <CRLF>	CHANGE TO PARENT DIRECTORY	Перейти к родительскому каталогу (удаленной файловой системы)
SMNT <SP> <pathname> <CRLF>	STRUCTURE MOUNT	Смонтировать требующуюся файловую систему
REIN <CRLF>	REINITIALIZE	Повторно инициализировать (пользователь выводится из удаленной системы с завершением текущего действия и сохранением настроек)
QUIT <CRLF>	LOGOUT	Выход из удаленной системы (с завершением текущего действия и закрытием управляющего соединения)
FEAT <CRLF>	FEATURE	Предоставить информацию о поддерживаемых расширениях (ключевые слова в FTP-ответе) (RFC 2389)

FTP-команды

10.1.2.2b

Transfer parameter commands (команды управления пересылкой)		
PORT <SP> <host-port> <CRLF>	DATA PORT	Совокупность IP-адреса и номера порта, необходимая для создания информационного соединения (<host-port> пересыпается в виде: <h1> "," <h2> "," <h3> "," <h4> "," <p1> "," <p2>; где <h1> ... <h4> – разбитый на четыре разделенных запятыми байта IP-адрес в десятичном представлении, причем байты следуют в правильном порядке; <p1> и <p2> – аналогичным образом разбитый номер порта, причем байты так же следуют начиная со старшего)
PASV <CRLF>	PASSIVE	Установить пассивный режим обмена
TYPE <SP> <type-code> <CRLF>	REPRESENTATION TYPE	Файловое представление (коды: а – ASCII, н – non-print, т – Telnet format effectors, е – EBCDIC, с –carriage control, и – image, л – local byte size, по умолчанию: а и н)
STRU <SP> <structure-code> <CRLF>	FILE STRUCTURE	Структура (коды: ф – file structure, р – record structure, п – page structure, по умолчанию: ф)
MODE <SP> <mode-code> <CRLF>	TRANSFER MODE	Режим пересылки (коды: с – stream, в – block, з – compressed, по умолчанию: с)
OPTS <SP> <command-name> [<SP> <command-options>] <CRLF>	OPTIONS	Задать опции обработки FTP-команды при ее последующих вызовах (если опции предусмотрены) (RFC 2389)
EPRT <SP> " " <net-prt> " " <net-addr> " " <tcp-port> " " <CRLF>	EXTENDED PORT	Совокупность номера семейства протоколов, IP-адреса и номера порта ... (расширенный вариант PORT; <net-prt>: 1 – IPv4, 2 – IPv6; <net-addr> и <tcp-port> пересыпаются в стандартной нотации) (RFC 2428)
EPSV [<SP> (<net-prt> ALL)] <CRLF>	EXTENDED PASSIVE MODE	Установить пассивный режим обмена (расширенный вариант PASV, может быть указан порт, аргумент ALL позволяет отменить действие команды) (RFC 2428)

FTP-команды

10.1.2.2c

Service commands (сервисные команды)		
RETR <SP> <pathname> <CRLF>	RETRIEVE	Загрузить файл с FTP-сервера (download)
STOR <SP> <pathname> <CRLF>	STORE	Загрузить файл на FTP-сервер (upload, если файл уже существует, то он обновляется)
STOU <CRLF>	STORE UNIQUE	Загрузить файл на FTP-сервер и сохранить там под уникальным названием
APPE <SP> <pathname> <CRLF>	APPEND (with create)	Загрузить файл на FTP-сервер с дозаписью (если файл уже существует, то данные дописываются в его конец)
ALLO <SP> <decimal-integer> [<SP> R <SP> <decimal-integer>] <CRLF>	ALLOCATE	Зарезервировать на FTP-сервере файловое пространство (в байтах, причем, если планируемый для пересылки файл имеет структуру R либо r, то после символа R указывается максимальный размер записи либо страницы)
REST <SP> <marker> <CRLF>	RESTART	Начать пересылку файла с указанного смещения в нем (используется для организации докачки) (+RFC 3659)
RNFR <SP> <pathname> <CRLF>	RENAME FROM	Старое название переназываемого файла на FTP-сервере (либо старый путь для локально переназываемого файла на FTP-сервере)
RNTO <SP> <pathname> <CRLF>	RENAME TO	Новое название переназываемого файла на FTP-сервере (должна следовать непосредственно за RNFR)
ABOR <CRLF>	ABORT	Принудительно завершить (предыдущую FTP-команду и связанную с ней пересылку)
DELE <SP> <pathname> <CRLF>	DELETE	Удалить файл либо каталог на FTP-сервере
RMD <SP> <pathname> <CRLF>	REMOVE DIRECTORY	Удалить каталог на FTP-сервере
MKD <SP> <pathname> <CRLF>	MAKE DIRECTORY	Создать каталог на FTP-сервере
PWD <CRLF>	PRINT WORKING DIRECTORY	Вывести на экран рабочий каталог
LIST [<SP> <pathname>] <CRLF>	LIST	Вывести на экран детализированный список файлов из удаленного каталога (если путь не указан, то подразумевается рабочий каталог)

FTP-команды

10.1.2.2d

<code>NLST [<SP> <pathname>] <CRLF></code>	NAME LIST	Вывести на экран упрощенный список файлов из удаленного каталога
<code>SITE <SP> <string> <CRLF></code>	SITE PARAMETERS	Предоставить специфическую системную информацию
<code>SYST <CRLF></code>	SYSTEM	Предоставить информацию об ОС сервера
<code>STAT [<SP> <pathname>] <CRLF></code>	STATUS	Предоставить информацию о текущем состоянии FTP-сервера или пересылки (FTP-ответ может пересыпаться как по управляемому, так и по информационному соединению)
<code>HELP [<SP> <string>] <CRLF></code>	HELP	Предоставить справочную информацию (обычно информацию о FTP-команде; если аргумент не задан, то выдается обобщенная справка)
<code>NOOP <CRLF></code>	NOOP	Холостая FTP-команда (обычно используется для поддержания связи)
<code>MDTM <SP> <pathname> <CRLF></code>	MODIFICATION TIME	Предоставить дату и время последней модификации файла (RFC 3659)
<code>SIZE <SP> <pathname> <CRLF></code>	SIZE OF FILE	Предоставить размер файла (RFC 3659)
<code>MLST <SP> <pathname> <CRLF></code>	--	Предоставить информацию об объекте файловой системы (файле либо каталоге; если путь не указан, то подразумевается рабочий каталог; FTP-ответ пересыпается по управляемому соединению) (RFC 3659)
<code>MLSD <SP> <pathname> <CRLF></code>	--	Вывести на экран список файлов из удаленного каталога (более стандартизированная альтернатива <code>LIST</code> (RFC 3659))

FTP-команды

10.1.2.3

В расширениях FTP есть еще несколько новых FTP-команд.

Около десяти FTP-команд оказались невостребованными и были аннулированы.

10.1.3.1

Каждая FTP-команда, переданная FTP-клиентом, должна сопровождаться по крайней мере одним FTP-ответом со стороны FTP-сервера, сообщающим об успешности ее выполнения.

В нормальной ситуации, FTP-клиент ожидает FTP-ответ на текущую FTP-команду перед тем, как передать следующую. При этом **используется** механизм тайм-аута.

В зависимости от реализации, на часть FTP-команд могут возвращаться различные комбинации FTP-ответов, однако существуют и жесткие ограничения.

Существуют также рекомендации по наполнению FTP-ответов текстом.

10.1.3.2

FTP-ответ, состоящий из одной строки, формально **выглядит** следующим образом:

```
xuz <SP> <text> <CRLF>
```

Где: xuz -- целочисленный трехбайтный код.

Если же FTP-ответ состоит из нескольких строк, что так же **допустимо**, он выглядит:

```
xuz " - " <text> <CRLF>
<text> <CRLF>
...
xuz <text> <CRLF>
```

Коды предназначены для техники, а текстовые комментарии -- для людей.

10.1.3.3а

Код	Название	Описание
1yz	Positive preliminary reply	Предварительное успешное завершение
2yz	Positive completion reply	Окончательное успешное завершение
3yz	Positive intermediate reply	Промежуточное успешное завершение
4yz	Transient negative completion reply	Ненормальное завершение в текущем случае
5yz	Permanent negative completion reply	Перманентно ненормальное завершение
x0z	Syntax	Синтаксис
x1z	Information	Информация
x2z	Connections	Соединения
x3z	Authentication and accounting	Аутентификация и аккаунты
x4z	Unspecified as yet	Еще стандартом не определено
x5z	File system	Файловая система

Декодирование FTP-ответов

10.1.3.3b

110	Restart marker reply	Подтверждение изменения файлового смещения (должно быть в формате: MARK <SP> <уууу> <SP> "=" <SP> <ттттт>, где <уууу> и <ттттт> -- файловые смещения на сторонах процессов пересылки данных FTP-клиента и FTP-сервера соответственно)
120	Service ready in <ппп> minutes	Запрос планируется обслужить за <ппп> минут
125	Data connection already open; transfer starting	Информационное соединение установлено и пересылка начинается
150	File status okay; about to open data connection	Файл корректен, подготавливается информационное соединение
200	Command okay	Команда выполнена успешно
202	Command not implemented, superfluous at this site	В выполнении команды нет необходимости
211	System status, or system help reply	Состояние системы или справка
212	Directory status	Состояние каталога
213	File status	Состояние файла
214	Help message	Справочное сообщение
215	<NAME> system type	Официальный тип системы: <NAME>
220	Service ready for new user	Готовность обслуживать нового пользователя (обычно содержит баннер)
221	Service closing control connection	Управляющее соединение закрывается
225	Data connection open	Информационное соединение установлено
226	Closing data connection	Информационное соединение закрывается
227	Entering Passive Mode	Пассивный режим обмена установлен
230	User logged in, proceed	Пользователь вошел в систему, можно продолжать
250	Requested file action okay, completed	Запрошенное действие с файлом выполнено
257	<PATHNAME> created	Файл <PATHNAME> создан
331	User name okay, need password	Имя пользователя воспринято, требуется пароль
332	Need account for login	Требуется аккаунт для входа в систему
350	Requested file action pending further information	Запрошенное действие с файлом отложено до поступления дополнительной информации

Декодирование FTP-ответов

10.1.3.3с

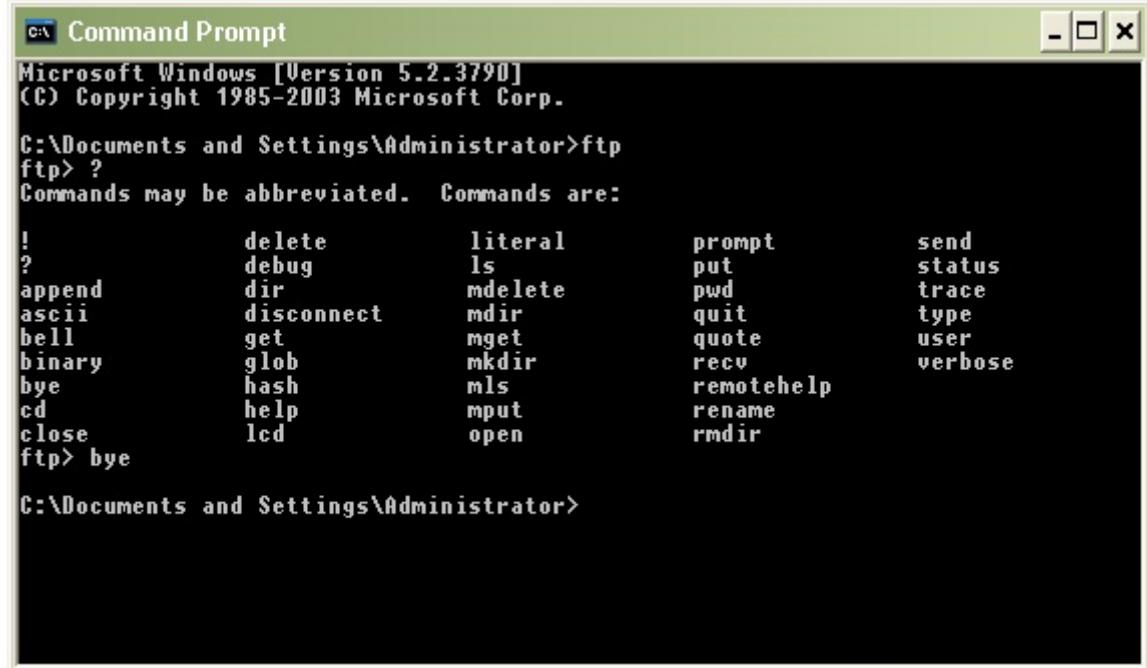
421	Service not available, closing control connection	Сервис недоступен, управляющее соединение закрывается
425	Can't open data connection	Невозможно установить информационное соединение
426	Connection closed; transfer aborted	Соединение закрыто, пересылка прервана
450	Requested file action not taken	Запрошенное действие с файлом не выполнено
451	Requested action aborted: local error in processing	Запрошенное действие прервано, локальная ошибка при обработке
452	Requested action not taken. Insufficient storage space in system	Запрошенное действие не выполнено, недостаточно свободного файлового пространства в системе
500	Syntax error, command unrecognized	Синтаксическая ошибка, команда не распознана
501	Syntax error in parameters or arguments	Синтаксическая ошибка в параметрах или аргументах
502	Command not implemented	Команда не выполнена
503	Bad sequence of commands	Неправильная последовательность команд
504	Command not implemented for that parameter	Команда с этим параметром не реализована
522	Network protocol not supported	Сетевой протокол не поддерживается (RFC 2428)
530	Not logged in	Вход в систему не осуществлен
532	Need account for storing files	Требуется аккаунт для сохранения файла
550	Requested action not taken. File unavailable	Запрошенное действие не выполнено, файл недоступен
551	Requested action aborted: page type unknown	Запрошенное действие прервано, тип страницы неизвестен
552	Requested file action aborted. Exceeded storage allocation	Запрошенное действие с файлом прервано, пространство на накопителе не выделено
553	Requested action not taken. File name not allowed	Запрошенное действие не выполнено, название файла недопустимо

Декодирование FTP-ответов

10.1.4.1

В отличие от FTP-команд, команды интерпретатора не стандартизированы, однако в реализациях широко **используют** традиционно сложившиеся аббревиатуры.

10.1.4.2



Command Prompt
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.
C:\Documents and Settings\Administrator>ftp
ftp> ?
Commands may be abbreviated. Commands are:
! delete literal prompt send
? debug ls put status
append dir mdelete pwd trace
ascii disconnect mdir quit type
bell get mget quote user
binary glob mkdir recv verbose
bye hash mls remotehelp
cd help mput rename
close lcd open rmdir
ftp> bye
C:\Documents and Settings\Administrator>

Команды интерпретатора FTP-клиента Windows

10.1.5.1

Протокол FTP разрабатывался как универсальный -- в том числе, и для пересылки файлов между станциями, работающими под управлением различных ОС, возможно использующих различные файловые системы. Для того чтобы обмен по протоколу прошел успешно, необходимо правильно задать или изменить используемые по умолчанию значения следующих параметров.

10.1.5.2

Файловое представление (data type, representation type) необходимо для согласования файловых систем передающей и принимающей сторон.

FTP поддерживает четыре основных файловых представления:

1. ASCII -- файл считается текстовым и пересыпается в 7-мибитной кодировке NVT-ASCII (по умолчанию).

2. EBCDIC -- файл считается текстовым и пересыпается в 8-мибитной кодировке EBCDIC фирмы IBM.

3. Image -- файл считается бинарным и пересыпается упакованным в 8-мибитные байты.

4. Local byte size -- файл считается состоящим из неделимых байтов соответствующего размера (должен быть задан) и пересыпается с учетом этого (если размер байта не кратен октету, то возникает автодополнение).

10.1.5.3

В дополнение к файловому представлению, в FTP предусмотрена возможность структурировать файл при его пересылке по информационному соединению (data structure).

Поддерживаются три структуры:

1. File structure -- файл не имеет внутренней структуры и рассматривается как непрерывный поток байтов (по умолчанию).
2. Record structure -- файл рассматривается как последовательность записей, структура приемлема только для текстовых файлов.
3. Page structure -- файл имеет страничную организацию, каждая страница имеет заголовок и индексацию, структура зависит от реализации.

10.1.5.4

Наконец, существует возможность установить режим пересылки (transmission mode).

Возможны три режима:

1. Stream -- файл пересыпается как непрерывный поток байтов (по умолчанию); если файл не имеет внутренней структуры, то прием метасимвола <EOF> означает, что пересылка окончена; для случаев со сложной структурой предусмотрены специальные коды для <EOR> и <EOF>.

2. Block -- файл пересыпается в виде последовательности блоков, каждый из которых имеет заголовок, в котором записываются счетчик байтов и специальные коды; способ поддерживается редко.

3. Compressed -- файл пересыпается в сжатом простейшими алгоритмами виде; способ поддерживается редко.

10.1.6.1

В зависимости от того, какая из взаимодействующих сторон является инициатором установления информационного соединения различают активный и пассивный режимы обмена (data transfer process modes).

При этом направление пересылки файлов, то есть какая из сторон является **отправителем**, а какая **получателем**, значения не имеет.

10.1.6.2

Активный режим является рекомендуемым и наиболее используемым. В активном режиме управляющее соединение создается следующим образом.

FTP-клиент, используя динамически выделенный порт (с номером больше 1024), создает управляющее соединение с портом 21 FTP-сервера.

Затем FTP-клиент динамически выделяет еще один порт и посыпает его номер FTP-серверу с помощью FTP-команды PORT.

Затем FTP-сервер создает информационное соединение с указанным портом, со своей стороны используя порт 20.

Если FTP-клиент не передал команду PORT, что в крайней степени не рекомендуется, то FTP-сервер создает информационное соединение с тем же самым портом FTP-клиента, который используется управляющим соединением.

10.1.6.3

Пассивный режим обычно устанавливается принудительно.

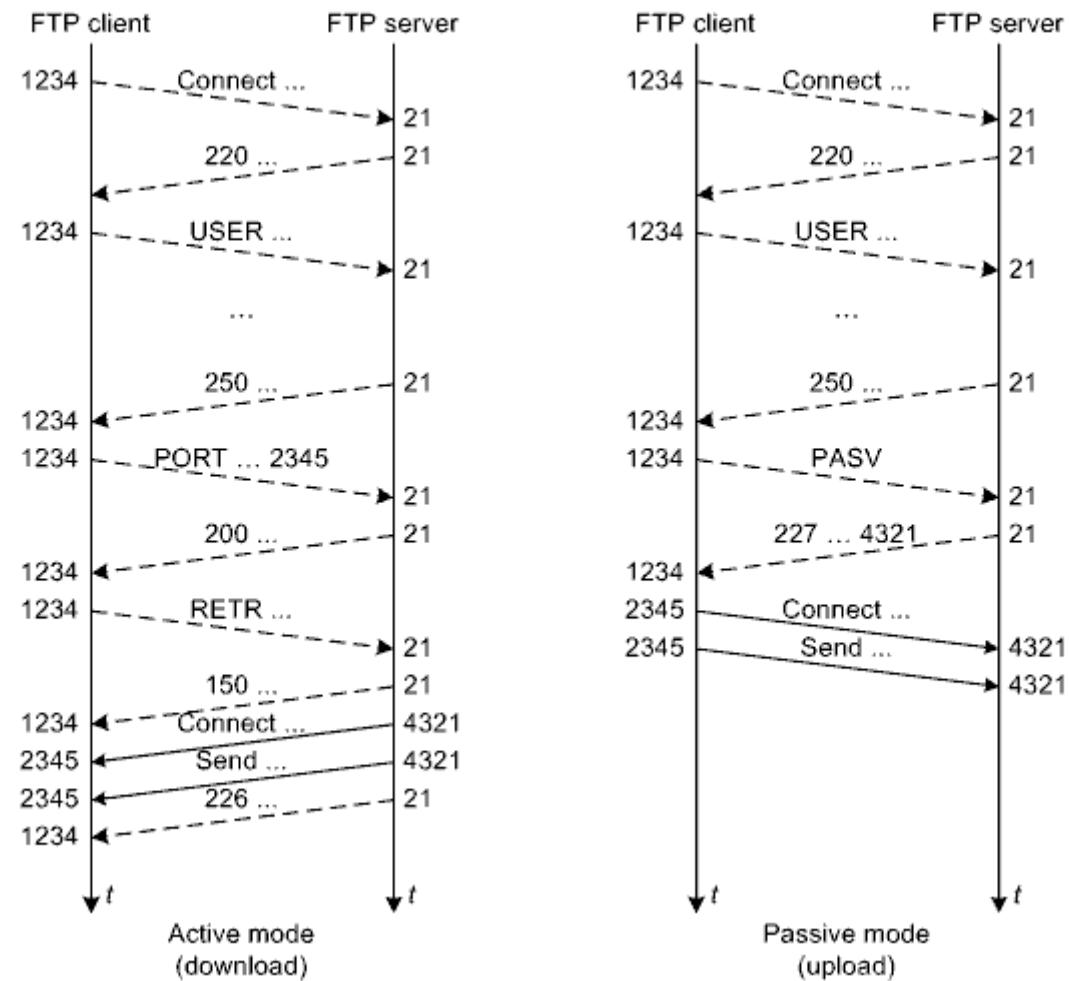
В пассивном режиме FTP-клиент создает как управляющее, так и информационное соединения с FTP-сервером.

После того, как аналогичным образом создано управляющее соединение, FTP-клиент передает FTP-команду PASV FTP-серверу.

Получив ее, FTP-сервер динамически выделяет порт для информационного соединения и передает его номер FTP-клиенту с помощью FTP-ответа 227.

Затем FTP-клиент динамически выделяет еще один порт и создает информационное соединение с портом, номер которого получил от FTP-сервера.

10.1.6.4



Режимы обмена по протоколу FTP

10.1.7.1

Широкое распространение получили так называемые «канонимные» (anonymous) FTP-серверы (RFC 1635), предоставляющие всем желающим определенные файловые ресурсы (обычно расположенные в каталоге /pub).

Как правило для аутентификации в такой системе достаточно ввести имя пользователя anonymous и произвольный пароль, например адрес электронной почты.

10.1.7.2

FTP имеет немного расширений, и те реализуют редко. Даже связанные с безопасностью расширения почти не **используют** (реализации базового стандарта совсем незащищены).

10.1.8.1

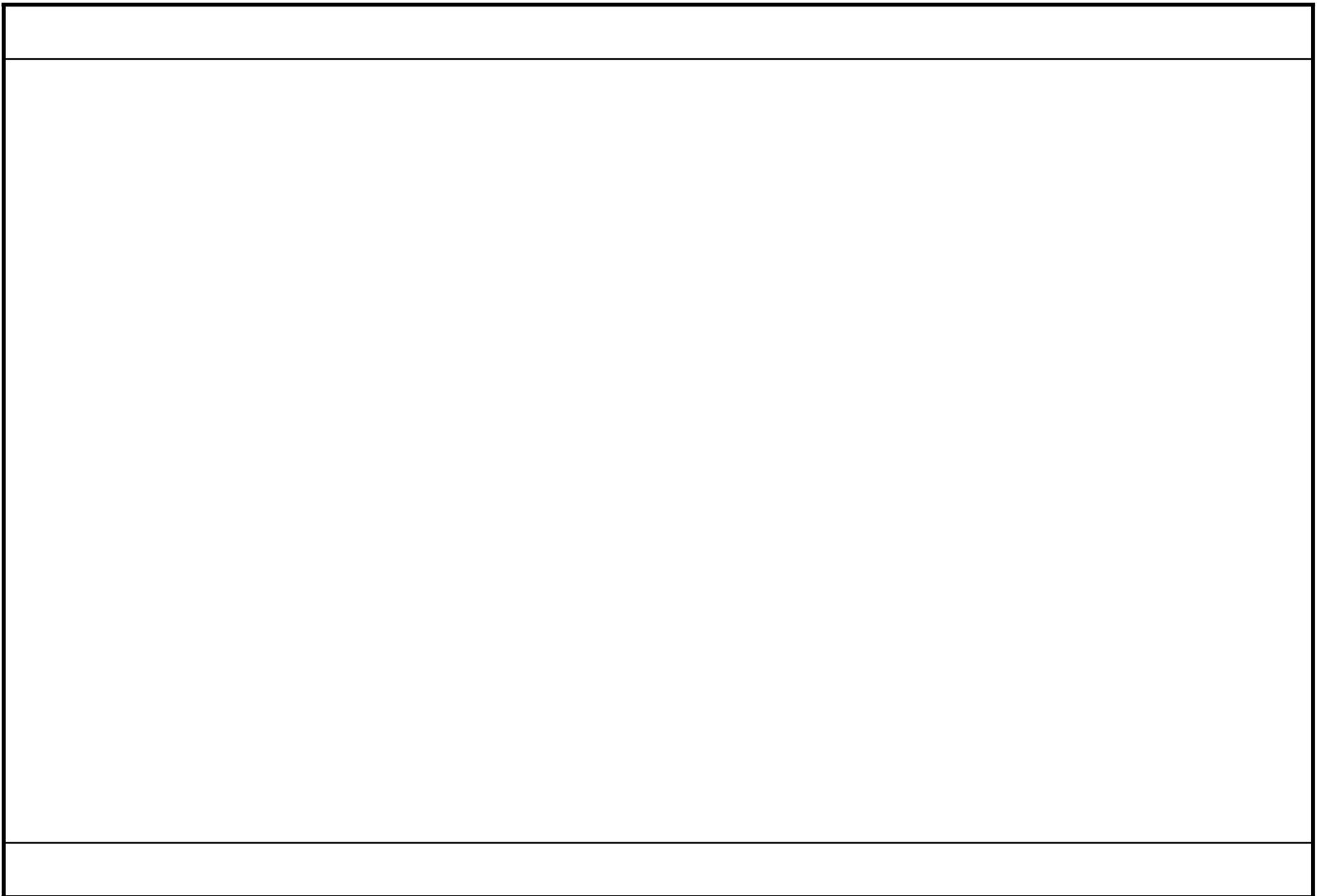
N	Дата	Время	Информация
1	2.2.2005	20:20:02	Состояние закачки - [Закачано]
2	2.2.2005	20:20:10	Состояние закачки - [Ожидание в очереди]
3	2.2.2005	20:20:11	Еще одна секция запущена
4	2.2.2005	20:20:11	Соединяемся с 192.168.11.2 (192.168.11.2:21)
5	2.2.2005	20:20:11	220 sunbasnet FTP server ready.
6	2.2.2005	20:20:11	USER ****
7	2.2.2005	20:20:11	331 Password required for user2.
8	2.2.2005	20:20:11	PASS ****
9	2.2.2005	20:20:11	230 User user2 logged in.
10	2.2.2005	20:20:11	SYST
11	2.2.2005	20:20:11	215 UNIX Type: L8 Version: SUNOS
12	2.2.2005	20:20:11	TYPE I
13	2.2.2005	20:20:11	200 Type set to I.
14	2.2.2005	20:20:11	REST 100
15	2.2.2005	20:20:11	350 Restarting at 100. Send STORE or RETRIEVE to initiate transfer.
16	2.2.2005	20:20:11	REST 0
17	2.2.2005	20:20:11	350 Restarting at 0. Send STORE or RETRIEVE to initiate transfer.
18	2.2.2005	20:20:11	PWD
19	2.2.2005	20:20:11	257 "/home2/user2" is current directory.
20	2.2.2005	20:20:11	CWD /home2/user2/RFC959.htm
21	2.2.2005	20:20:11	550 /home2/user2/RFC959.htm: Not a directory.
22	2.2.2005	20:20:11	PORT 192,168,11,22,4,56
23	2.2.2005	20:20:11	200 PORT command successful.
24	2.2.2005	20:20:11	LIST -la RFC959.htm
25	2.2.2005	20:20:11	150 Opening BINARY mode data connection for /bin/ls.
26	2.2.2005	20:20:11	Соединение установлено
27	2.2.2005	20:20:11	-rw-r--r-- 1 user2 users 145512 Feb 2 20:11 RFC959.htm
28	2.2.2005	20:20:11	226 Transfer complete.
29	2.2.2005	20:20:11	PORT 192,168,11,22,4,57
30	2.2.2005	20:20:11	200 PORT command successful.
31	2.2.2005	20:20:11	RETR RFC959.htm
32	2.2.2005	20:20:11	150 Opening BINARY mode data connection for RFC959.htm (145512 bytes).
33	2.2.2005	20:20:11	Соединение установлено
34	2.2.2005	20:20:11	Состояние закачки - [Закачка]
35	2.2.2005	20:20:12	226 Transfer complete.
36	2.2.2005	20:20:12	Секция скачана
37	2.2.2005	20:20:12	Состояние закачки - [Закачано]

Пример лога загрузки файла с FTP-сервера в активном режиме

10.1.8.2

▲	19	2.2.2005	20:23:55	CWD /home2/user2/RFC959.htm
▼	20	2.2.2005	20:23:55	550 ./home2/user2/RFC959.htm: Not a directory.
▲	21	2.2.2005	20:23:55	PASV
▼	22	2.2.2005	20:23:55	227 Entering Passive Mode (192,168,11,2,101,23)
▲	23	2.2.2005	20:23:55	LIST -la RFC959.htm
■	24	2.2.2005	20:23:55	Connecting to (192.168.11.2:25879)
▼	25	2.2.2005	20:23:55	150 Opening BINARY mode data connection for /bin/ls.
▼	26	2.2.2005	20:23:55	-rw-r--r-- 1 user2 users 145512 Feb 2 20:11 RFC959.htm
▼	27	2.2.2005	20:23:55	226 Transfer complete.

Отличающийся фрагмент примера лога загрузки в пассивном режиме



10.3

ПРОТОКОЛЫ ЭЛЕКТРОННОЙ ПОЧТЫ

Версия 2.5

10.3.1.1

Сообщениями протоколов электронной почты являются **электронные письма (emails)**.

Электронные письма имеют текстовую природу.

По аналогии с бумажным письмом, электронное письмо так же состоит из конверта (*envelope*) и содержимого (*content*).

Содержимое, в свою очередь, состоит из заголовка (*header*) и основного текста (*body*).

Структура (прежде всего, синтаксис) электронных писем неоднократно регламентировалась стандартами (начиная с RFC 822 и более старых, заканчивая RFC 5322).

Для обеспечения прав и обязанностей, связанных с электронными письмами, предусмотрены два механизма: DKIM (DomainKeys Identified Mail) Signatures (RFC 6376) и SPF (Sender Policy Framework) (RFC 7208).

Изначально, в отношении всех компонентов электронного письма допускалась только 7-мибитная кодировка US-ASCII.

10.3.1.2

Очень значимым расширением электронной почты является MIME (Multipurpose Internet Mail Extensions) (RFC 2045 -- 2049 и много обновлений), позволяющее включать в основной текст электронного письма (и «прикреплять» к электронному письму) различные мультимедийные данные.

В настоящее время определены следующие MIME-типы:

1. `text` -- текст (основное содержимое письма, указывается кодировка, возможна 8-мибитная кодировка).
2. `image` -- изображение.
3. `audio` -- звук.
4. `video` -- видео.
5. `application` -- электронные данные, не подпадающие ни под один из других типов.
 - +6. `multipart` -- комбинация нескольких типов.
 - +7. `message` -- письмо в письме либо внешнее приложение к письму (`attachment`).

Каждый из типов имеет некоторое количество подтипов.

10.3.1.3

Одно из последних расширений (RFC 6532) позволяет полностью интернационализировать электронную почту -- разрешает использовать кодировку UTF-8 вместо US-ASCII уже в адресах и напрямую в заголовках (второй шаг после MIME).

10.3.1.4

Стандарты четко не ограничивают размеры электронного письма и его составных частей.

Однако, в реализациях, размер содержимого обычно не должен превышать 64 килобайта, а общий объем (включая приложения) -- несколько мегабайтов.

10.3.1.5

Одним из ключевых понятий системы электронной почты является понятие почтового ящика (mailbox, иногда maildrop). Электронный почтовый ящик по своей сути ничем не отличается от почтового ящика для бумажных писем, но не всегда представляет собой файловое хранилище.

Почтовые ящики могут быть расположены как на выделенных для этого почтовых серверах -- лучше переводить как MXes (Mail eXchanges), так и на пользовательских станциях.

Могут быть как локальными, так и удаленными от пользователя.

10.3.1.6а

Фактически, в системе электронной почты **адресуют** именно почтовые ящики.

Современный формат адреса:

```
<address> = <mailbox> / <group>

<mailbox> = <name-addr> / <addr-spec>
<name-addr> = [<display-name>] [<CFWS>] "<" <addr-spec> ">" [<CFWS>] / <obs-angle-addr>
<addr-spec> = <local-part> @ <domain>

<group> = <display-name> ":" [<mailbox-list> / <CFWS> / <obs-group-list>] ";" [<CFWS>]
<mailbox-list> = (<mailbox> *("," <mailbox>)) / <obs-mbox-list>
```

Имеется совместимость с устаревшим форматом (не отображено).
Поддерживаются псевдонимы и групповая рассылка.

10.3.1.6b

Где:

<display-name> -- имя человека либо название программы (отправителя либо получателя, просто отображается, при пересылке не используется);

<local-part> -- название почтового ящика (отправителя либо получателя; может содержать все печатные US-ASCII-символы кроме метасимволов: (,), <, >, [,], :, ;, @, \, , "; метасимвол-точка является исключением, но точка не может быть крайней и не может встречаться более одного раза подряд; регистр букв учитывается; экранирование заключением в двойные кавычки позволяет включить в название все символы, при этом \ и " заменяются на \\ и \\" соответственно; максимум 64 байта);

<domain> -- доменное название, относящееся к домену либо к станции (источника либо назначения), где расположен данный почтовый ящик;

<CFWS> -- комментарий (в круглых скобках) или стандартный разделитель (один либо несколько пробелов или табуляций).

10.3.1.6c

Примеры:

John Doe <jdoe@machine.example>

"Mary Smith: Personal Account" <smith@home.example>

A Group(Some people)

:Chris Jones <c@(Chris's host.)public.example>,

joe@example.org,

John <jdoe@one.test> (my dear friend);

10.3.1.7

При рассмотрении любой почтовой системы прежде всего **выделяют** взаимодействующие процессы, которые принято называть почтовыми агентами.

Традиционно в список почтовых агентов включают:

1. MTAs (Mail Transport Agents) -- доставляют письма между почтовыми серверами.

2. MDAs (Mail Delivery Agents) -- помещают доставленные сообщения в почтовые ящики пользователей.

3. MUAs (Mail User Agents) -- реализуют интерфейс пользователей с их почтовыми ящиками.

+4. MSAs (Mail Submission Agents) -- позволяют вводить письма разными способами.

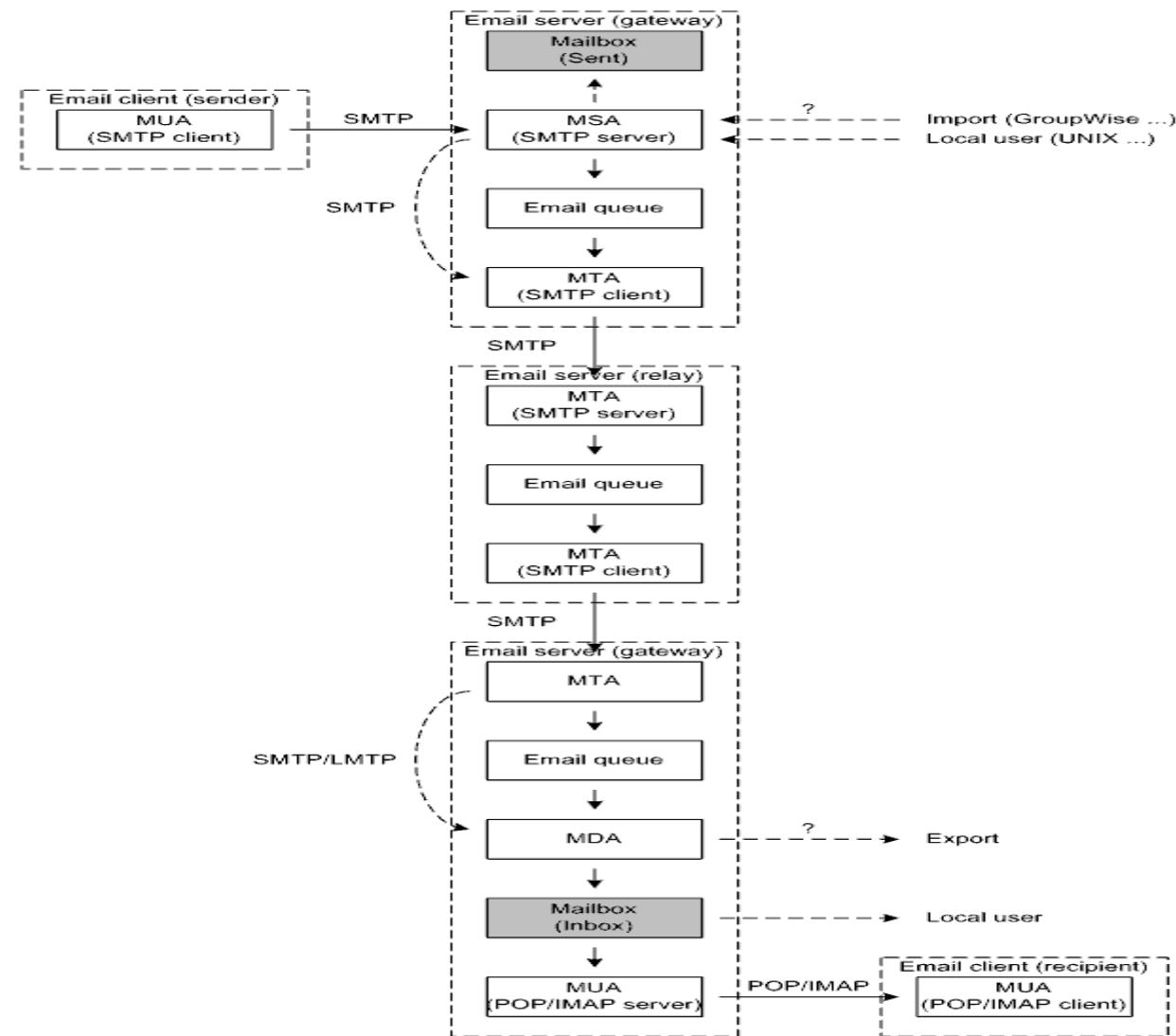
Следует отметить, что количественное и качественное наполнение списка почтовых агентов разнится (выше показан «усредненный» список). Например, в стандартах RFC термин MDA почти не **упоминают**. Изначально **выделяли** только MUA и MTA. Недавно на стороне отправителя между MUA и MTA был «вставлен» MSA.

10.3.1.8

Наличие удаленных почтовых ящиков (расположенных на удаленных от пользователя почтовых серверах) привело к такой отличительной особенности системы электронной почты как выделение в рамках одной прикладной задачи двух групп базовых прикладных протоколов -- протоколов для передачи почты и протоколов для приема почты.

В свою очередь, в рамках каждого из почтовых протоколов **взята** за основу клиент-серверная модель и используется транспорт TCP.

10.3.1.9



Обобщенная структура системы электронной почты

10.3.1.10

Чем отличается программа -- почтовый клиент от web-интерфейса для работы с почтовым ящиком?

10.3.2.1

Протокол SMTP (Simple Mail Transfer Protocol) (основное RFC -- RFC 5321) **используют** для передачи электронной почты в почтовый ящик -- от пользовательской станции (отправителя) к почтовому серверу и от одного почтового сервера к другому. (В очень редких случаях, при «необычных» почтовых ящиках, письмо может передаваться и от почтового сервера к пользовательской станции, например, «прямо на экран монитора»).

Таким образом, SMTP-клиентом (передатчиком письма) может быть как пользовательская станция, так и почтовый сервер, а SMTP-сервером (приемником письма) как правило является почтовый сервер.

Правила взаимодействия между SMTP-клиентом и SMTP-сервером аналогичны правилам взаимодействия между FTP-клиентом и FTP-сервером.

Задействуется одно соединение.

Стандартный номер программного порта SMTP-сервера -- 25.

Для MSA предусмотрен альтернативный номер порта -- 587, но многие почтовые системы для MSA по-прежнему используют порт 25.

10.3.2.2

Физическим почтовым клиентом является оконечная пользовательская станция, за которой работает пользователь-отправитель или пользователь-получатель письма.

Физическим почтовым сервером обычно является оконечная серверная станция.

Логическим почтовым клиентом и логическим почтовым сервером являются определенные программы.

Логический почтовый клиент не всегда находится на физическом почтовом клиенте, он может находиться на физическом почтовом сервере (что свойственно ОС UNIX, локальный пользователь может быть подключен и через удаленный терминал посредством SSH).

И почтовые агенты могут соотноситься со станциями по-разному.

На почтовом клиенте может располагаться только MUA, на почтовом сервере могут располагаться все почтовые агенты.

MSA плюс MTA, равно как MTA плюс MDA, обычно совмещены на одном почтовом сервере, но в общем случае могут быть «разнесены» (при этом взаимодействуют так же по протоколу SMTP).

10.3.2.3а

Согласно типовой схеме пересылки письма сначала отрабатывает связка MUA -- MSA.

MUA позволяет ввести письмо явно, либо указать MSA откуда взять письмо.

Опционально, отправляемое письмо копируется в соответствующую папку почтового ящика отправителя.

Затем MSA «ставит» письмо в очередь, отслеживаемую соответствующим МТА (если почтовый ящик получателя расположен на том же почтовом сервере, то письмо сразу «сбрасывается» в него).

Забрав письмо из очереди, МТА считывает <domain>-часть адреса почтового ящика получателя (MTAs анализируют только эту часть) и пытается установить соединение с MX домена -- почтовым сервером, отвечающим за указанный в адресе домен. Ведь заранее неизвестно, к домену либо непосредственно к станции относится считанное доменное название (исторически принята такая, косвенная, адресация через MX домена).

10.3.2.3b

Для поиска MX домена происходит обращение к системе DNS и используются записи о ресурсах типа MX (кроме «обычных» DNS-преобразований, которые никто не отменял).

Если почтовый ящик получателя не расположен на MX домена, то MX домена «продвигает» письмо дальше, то есть становится посредником (SMTP Relay). Таким образом, при пересылке письма по СПД от отправителя к получателю может возникать своеобразная маршрутизация.

Когда письмо «дошло» до станции назначения последний МТА в цепочке отдает его соответствующему MDA.

MDA анализирует <local-part>-часть адреса почтового ящика, находит почтовый ящик и помещает письмо в него.

Дальнейшая обработка письма «находится вне компетенции» SMTP.

10.3.2.3с

Очевидно, что при пересылке письма оно много раз буферизируется. Кроме специальных разделяемых очередей, имеются и буфера в почтовых агентах. Очередь можно рассматривать как средство межпроцессного взаимодействия. Очередь так же позволяет предпринять повторную попытку передать письмо после промежуточного сбоя.

Наличие MSA позволяет импортировать письма в том числе из корпоративных почтовых систем (Novell GroupWise, Lotus Notes -- IBM Domino, Microsoft Exchange и других), которые не полностью соответствуют стандартам RFC. Наличие MDA позволяет экспортить письма. Выполняющий импорт-экспорт MX становится почтовым шлюзом (email gateway) между различными почтовыми системами. При пересылке письма MIME-типы могут преобразовываться.

Для локальной доставки от MTA к MDA иногда используется специально для этого разработанная модификация SMTP под названием LMTP (Local Mail Transfer Protocol) (RFC 2033).

10.3.2.4

SMTP имеет целый ряд расширений, обобщенно известных как ESMTP (Extended SMTP), многие из которых уже интегрированы в основной стандарт.

Некоторые расширения зависят друг от друга.

SMTP поддерживает MIME.

Аутентификация SMTP-клиента на SMTP-сервере выполняется посредством механизма SASL (Simple Authentication and Security Layer) (изначально аутентификация предусмотрена не была).

Еще одним важным опциональным расширением, связанным с безопасностью, является механизм TLS (Transport Layer Security) -- новое название SSL (Secure Sockets Layer), позволяющий защитить соединение от несанкционированного доступа.

Опционально SMTP поддерживает конвейеризацию команд (pipelining), то есть способность отправлять текущую SMTP-команду по мере необходимости, возможно еще не получив SMTP-ответ на предыдущую.

SMTP-клиент отслеживает различные тайм-ауты.

10.3.3.1

Формат SMTP-команд аналогичен формату FTP-команд.
SMTP-команда начинается с четырехбуквенного кода -- по-другому,
глагола (verb).

Регистр букв не учитывается.

10.3.3.2а

SMTP-команда	Название	Описание
<code>HELO <SP> <Domain> <CRLF></code>	HELLO	Идентифицировать на SMTP-сервере (под своим доменным названием)
<code>EHLO <SP> (<Domain> / <address-literal>) <CRLF></code>	EXTENDED HELLO	Идентифицировать на SMTP-сервере для использования ESMTP (рекомендуемая альтернатива HELO)
<code>MAIL FROM: <Reverse-path> [<SP> <Mail-parameters>] <CRLF></code>	MAIL	Почтовый ящик отправителя (MAIL FROM: инициирует транзакцию передачи письма) (+RFC 1845, +RFC 1870, +RFC 2852, +RFC 3030, +RFC 3461, +RFC 3865, +RFC 3885, +RFC 4141, +RFC 4405, +RFC 4865, +RFC 4954, +RFC 6152, +RFC 6531, +RFC 6710)
<code>RCPT TO: ("<Postmaster@<Domain>" ">" / "<Postmaster>" / <Forward-path>) [<SP> <Rcpt-parameters>] <CRLF></code>	RECIPIENT	Почтовый ящик получателя (RCPT TO: должна следовать непосредственно за MAIL FROM:; если получателей несколько, то эта команда повторяется соответствующее число раз) (+RFC 3461, +RFC 4141, +RFC 7293)
<code>DATA <CRLF></code>	DATA	Содержимое письма (DATA должна следовать непосредственно за RCPT TO:; следующие непосредственно за DATA <CRLF> строки рассматриваются как содержимое письма – вплоть до <CRLF>. <CRLF>)
<code>RSET <CRLF></code>	RESET	Отменить (текущую транзакцию посыпки письма)
<code>VRFY <SP> <String> [<SP> SMTPUTF8] <CRLF></code>	VERIFY	Верифицировать (что запрашиваемый объект существует и является почтовым ящиком) (+RFC 6531)
<code>EXPN <SP> <String> [<SP> SMTPUTF8] <CRLF></code>	EXPAND	Верифицировать список почтовой рассылки и, в положительном случае, предоставить список почтовых ящиков (+RFC 6531)
<code>HELP [<SP> <String>] <CRLF></code>	HELP	Предоставить справочную информацию
<code>NOOP [<SP> <String>] <CRLF></code>	NOOP	Холостая SMTP-команда
<code>QUIT <CRLF></code>	QUIT	Выход из удаленной системы

SMTP-команды

10.3.3.2b

<code>ETRN [<SP> <option character>] <SP> <node name> <CRLF></code>	--	Обработать очередь писем на SMTP-сервере, относящуюся к указанной станции (RFC 1985)
<code>BDAT <SP> <chunk-size> [<SP> LAST] <CRLF></code>	--	Фрагмент содержимого письма (при фрагментированной посыпке, альтернатива DATA) (RFC 3030)
<code>ATRN [<SP> <domain> * (", " <domain>)] <CRLF></code>	AUTHENTICATED TURN	Сменить роли SMTP-клиента и SMTP-сервера на охваченных соединением станциях (чтобы клиент стал сервером и наоборот; почти бессмысленно; возможно указание доменов, где выполнять смену) (RFC 2645)
<code>STARTTLS <CRLF></code>	START TLS	Запустить TLS (затем должно следовать согласование TLS-параметров) (RFC 3207)
<code>BURL <SP> <absolute-URI> [<SP> LAST] <CRLF></code>	--	Ссылка на содержимое письма (при импорте из IMAP, альтернатива DATA) (RFC 4468)
<code>AUTH <SP> <mechanism> [<SP> <initial-response>] <CRLF></code>	AUTHENTICATION	Аутентифицировать в соответствии с указанным механизмом SASL (затем должна следовать аутентификация) (RFC 4954)
<code>X ... <CRLF></code>	--	Выполнить «нестандартную» SMTP-команду (признаком является первая буква X в названии, аргументы не регламентируются)

SMTP-команды

10.3.3.3

Таким образом, SMTP-команды MAIL FROM:, RCPT TO: и DATA -- это три шага транзакции передачи письма.

Первые два шага образуют конверт письма, а третий шаг -- это передача содержимого письма с небольшим обрамлением.

Некоторые SMTP-команды (SEND, SOML, SAML, TURN) в настоящее время аннулированы.

10.3.4.1

Базовая схема декодирования SMTP-ответов аналогична схеме декодирования FTP-ответов.

10.3.4.2а

Код	Название	Описание
2yz	Positive completion reply	Окончательное успешное завершение
3yz	Positive intermediate reply	Промежуточное успешное завершение
4yz	Transient negative completion reply	Ненормальное завершение в текущем случае
5yz	Permanent negative completion reply	Перманентное ненормальное завершение
x0z	Syntax	Синтаксис
x1z	Information	Информация
x2z	Connections	Соединения
x3z	Unspecified	Не определено
x4z	Unspecified	Не определено
x5z	Mail system	Почтовая система

Декодирование базовых SMTP-ответов

10.3.4.2b

211	System status, or system help reply	Состояние системы или справка
214	Help message	Справочное сообщение (для людей)
220	<domain> Service ready	Сервис <domain> готов
221	<domain> Service closing transmission channel	Сервис <domain> закрывает канал пересылки
235	Authentication Succeeded	Аутентификация прошла успешно (RFC 4954)
250	Requested mail action okay, completed	Запрошенное действие, связанное с почтой, успешно завершено
251	User not local; will forward to <forward-path>	Пользователь не является локальным, перенаправление на <forward-path>
252	Cannot VRFY user, but will accept message and attempt delivery	Невозможно верифицировать пользователя, но сообщение будет принято и будет осуществлена попытка его доставить
354	Start mail input; end with <CRLF>.<CRLF>	Начинайте вводить письмо, ввод завершите <CRLF>.<CRLF>
421	<domain> Service not available, closing transmission channel	Сервис <domain> недоступен, канал пересылки закрывается
432	A password transition is needed	Требуется пересылка пароля (RFC 4954)
450	Requested mail action not taken: mailbox unavailable	Запрошенное действие с почтой не выполнено, почтовый ящик недоступен
450	ATRN request refused	Запрос ATRN отклонен (RFC 2645)
451	Requested action aborted: error in processing	Запрошенное действие прервано, обработка ошибки
451	Unable to process ATRN request now	В настоящий момент невозможно обработать запрос ATRN (RFC 2645)
452	Requested action not taken: insufficient system storage	Запрошенное действие с почтой не выполнено, недостаточно места на накопителе
453	You have no mail	Для вас почты нет (RFC 2645)
454	Temporary authentication failure	Временная проблема с аутентификацией (сбой на SMTP-сервере) (RFC 4954)
455	Server unable to accommodate parameters	Сервер не может подобрать параметры

Декодирование базовых SMTP-ответов

10.3.4.2с

500	Syntax error, command unrecognized	Синтаксическая ошибка, команда не распознана
500	Authentication Exchange line is too long	Слишком длинная строка при аутентификационном обмене (RFC 4954)
501	Syntax error in parameters or arguments	Синтаксическая ошибка в параметрах или аргументах
502	Command not implemented	Команда не реализована
503	Bad sequence of commands	Неправильная последовательность команд
504	Command parameter not implemented	Параметр команды не реализован
521	Server does not accept mail и другие варианты	Сервер не воспринял письмо (RFC 7504)
523	Text: Encryption Needed	Требуется шифрование текста (RFC 5248)
525	User Account Disabled	Пользовательский аккаунт запрещен (RFC 5248)
534	Authentication mechanism is too weak	Механизм аутентификации слишком уязвим (RFC 4954)
535	Authentication credentials invalid	Неправильные значения параметров аутентификации (RFC 4954)
538	Encryption required for requested authentication mechanism	Запрошенный механизм аутентификации предполагает шифрование (RFC 4954)
550	Requested action not taken: mailbox unavailable	Запрошенное действие не выполнено, почтовый ящик недоступен
551	User not local; please try <forward-path>	Пользователь не является локальным, попробуйте использовать <forward-path>
552	Requested mail action aborted: exceeded storage allocation	Запрошенное действие с почтой не выполнено, пространство на накопителе не выделено
553	Requested action not taken: mailbox name not allowed	Запрошенное действие с почтой не выполнено, недопустимое название почтового ящика
554	Transaction failed	Сбой транзакции
554	No SMTP service here	SMTP-сервис отсутствует
555	MAIL FROM/RCPT TO parameters not recognized or not implemented	Параметры MAIL FROM/RCPT TO не распознаны или не реализованы
556	Разные варианты	SMTP-сервер не воспринял письмо (альтернатива 521 при использовании почтовых систем-посредников, разные причины) (RFC 7504, RFC 7505)

Декодирование базовых SMTP-ответов

10.3.4.3

Новая, расширенная схема декодирования SMTP-ответов (RFC 3463 и RFC 5248) «открывает простор» для большей детализации и **рекомендована для использования** (предполагает дальнейшее наполнение).

10.3.4.4а

Код	Название	Описание
2.xxx.xxx	Success	Успешное завершение
4.xxx.xxx	Persistent Transient Failure	Устойчивый промежуточный сбой
5.xxx.xxx	Permanent Failure	Перманентный сбой
X.0.xxx	Other or Undefined Status	Прочее или неопределенное состояние
X.1.xxx	Addressing Status	Состояние подсистемы адресации
X.2.xxx	Mailbox Status	Состояние почтового ящика
X.3.xxx	Mail System Status	Состояние почтовой системы
X.4.xxx	Network and Routing Status	Состояние СПД
X.5.xxx	Mail Delivery Protocol Status	Состояние почтового протокола
X.6.xxx	Message Content or Media Status	Состояние письма
X.7.xxx	Security or Policy Status	Состояние подсистем об обеспечения безопасности и политики обслуживания

Декодирование расширенных SMTP-ответов

10.3.4.4b

x.0.0	Other undefined Status	Прочее неопределенное состояние
x.1.0	Other address status	Прочее состояние подсистемы адресации
x.1.1	Bad destination mailbox address	Неправильный адрес почтового ящика назначения
x.1.2	Bad destination system address	Неправильный адрес системы назначения
x.1.3	Bad destination mailbox address syntax	Синтаксически неправильный адрес почтового ящика назначения
x.1.4	Destination mailbox address ambiguous	Неоднозначный адрес почтового ящика назначения
x.1.5	Destination address valid	Правильный адрес назначения
x.1.6	Destination mailbox has moved, No forwarding address	Почтовый ящик назначения перемещен, нет адреса для пересылки
x.1.7	Bad sender's mailbox address syntax	Синтаксически неправильный адрес почтового ящика отправителя
x.1.8	Bad sender's system address	Неправильный адрес системы отправителя
x.1.9	Message relayed to non-compliant mailer	Письмо передано серверу-посреднику, который не удовлетворяет требованиям протокола (RFC 3886)
x.1.10	Recipient address has null MX	Домен, название которого указано в адресе получателя, не имеет MX (RFC 7505)
x.2.0	Other or undefined mailbox status	Прочее или неопределенное состояние почтового ящика
x.2.1	Mailbox disabled, not accepting messages	Работа с почтовым ящиком запрещена, письма не принимаются
x.2.2	Mailbox full	Почтовый ящик заполнен
x.2.3	Message length exceeds administrative limit	Размер письма превышает административно установленный лимит
x.2.4	Mailing list expansion problem	Проблема с интерпретацией списка почтовой рассылки

Декодирование расширенных SMTP-ответов

10.3.4.4c

X.3.0	Other or undefined mail system status	Прочее или неопределенное состояние почтовой системы
X.3.1	Mail system full	Накопитель почтовой системы заполнен
X.3.2	System not accepting network messages	Система не воспринимает сетевые сообщения (RFC 7504)
X.3.3	System not capable of selected features	Система не поддерживает выбранные возможности
X.3.4	Message too big for system	Письмо слишком большое для системы
X.3.5	System incorrectly configured	Система сконфигурирована некорректно
X.3.6	Requested priority was changed	Запрошенный приоритет был изменен (RFC 6710)
X.4.0	Other or undefined network or routing status	Прочее или неопределенное состояние СПД
X.4.1	No answer from host	Хост не отвечает
X.4.2	Bad connection	Неправильное соединение
X.4.3	Directory server failure	Сбой при обращении к системе DNS
X.4.4	Unable to route	Невозможно выполнить маршрутизацию
X.4.5	Mail system congestion	Затор в почтовой системе
X.4.6	Routing loop detected	Обнаружен маршрутационный цикл
X.4.7	Delivery time expired	Время доставки истекло
X.5.0	Other or undefined protocol status	Прочее или неопределенное состояние почтового протокола
X.5.1	Invalid command	Неправильная команда
X.5.2	Syntax error	Синтаксическая ошибка
X.5.3	Too many recipients	Слишком много потребителей почты
X.5.4	Invalid command arguments	Неправильные аргументы команды
X.5.5	Wrong protocol version	Несоответствующая версия протокола
X.5.6	Authentication Exchange line is too long	Слишком длинная строка при аутентификационном обмене (RFC 4954)

Декодирование расширенных SMTP-ответов

10.3.4.4d

x.6.0	Other or undefined media error	Прочее или неопределенное состояние письма
x.6.1	Media not supported	Формат письма не поддерживается
x.6.2	Conversion required and prohibited	Требуется преобразование письма, но преобразование запрещено
x.6.3	Conversion required but not supported	Требуется преобразование письма, но преобразование не поддерживается
x.6.4	Conversion with loss performed	Преобразование письма выполнено, но часть данных потеряна
x.6.5	Conversion Failed	Сбой при преобразовании письма
x.6.6	Message content not available	Содержимое письма недоступно (при ссылке на содержимое, которое находится на удаленном SMTP-сервере) (RFC 4468)
x.6.7	Non-ASCII addresses not permitted for that sender/recipient	Не-ASCII-адреса указанного отправителя-получателя не разрешены (RFC 6531)
x.6.8	UTF-8 string reply is required, but not permitted by the SMTP client	Ответ должен быть в кодировке UTF-8, но это запрещено SMTP-клиентом (RFC 6531)
x.6.9	UTF-8 header message cannot be transferred to one or more recipients, so the message must be rejected	Письмо с заголовком в кодировке UTF-8 не может быть переслано одному либо нескольким получателям, поэтому должно быть отброшено (RFC 6531)

Декодирование расширенных SMTP-ответов

10.3.4.4e

x.7.0	Other or undefined security status	Прочее или неопределенное состояние подсистем обеспечения безопасности и политики обслуживания
x.7.1	Delivery not authorized, message refused	Нет прав отсылать письма, письмо отброшено
x.7.2	Mailing list expansion prohibited	Интерпретация списков почтовой рассылки запрещена
x.7.3	Security conversion required but not possible	Требуется обеспечение защиты при преобразовании письма, но возможности обеспечить защиту нет
x.7.4	Security features not supported	Связанные с обеспечением защиты возможности не поддерживаются
x.7.5	Cryptographic failure	Сбой связан с криптографией
x.7.6	Cryptographic algorithm not supported	Криптографический алгоритм не поддерживается
x.7.7	Message integrity failure	Целостность письма нарушена
x.7.8	Authentication credentials invalid	Неправильные значения параметров аутентификации (RFC 4954)
x.7.9	Authentication mechanism is too weak	Механизм аутентификации слишком уязвим (RFC 4954)
x.7.10	Text: Encryption Needed	Требуется шифрование текста (RFC 5248)
x.7.11	Encryption required for requested authentication mechanism	Запрошенный механизм аутентификации предполагает шифрование (RFC 4954)
x.7.12	A password transition is needed	Требуется пересыпка пароля (RFC 4954)
x.7.13	User Account Disabled	Пользовательский аккаунт запрещен (RFC 5248)
x.7.14	Trust relationship required	Требуется установление доверительных отношений (между MSA и удаленным сервером для получения доступа к содержимому письма) (RFC 5248)
x.7.15	Priority Level is too low	Уровень приоритета слишком низкий (RFC 6710)
x.7.16	Message is too big for the specified priority	Письмо слишком велико чтобы обеспечить указанный приоритет (RFC 6710)
x.7.17	Mailbox owner has changed	Владелец почтового ящика сменился (RFC 7293)
x.7.18	Domain owner has changed	Владелец домена сменился (RFC 7293)
x.7.19	RRVS test cannot be completed	RRVS-тест невозможно выполнить (RFC 7293)
x.7.27	Sender address has null MX	Домен, название которого указано в адресе отправителя, не имеет MX (RFC 7505)

Декодирование расширенных SMTP-ответов

10.3.5.1а

В настоящее время в состав ESMTP входят следующие расширения (перечисляются в SMTP-ответе на SMTP-команду EHLO и клиент «делает выводы»).

SMTP-расширение	Описание
<code>EXPN</code>	Верификация списков почтовой рассылки
<code>HELP</code>	На сервере имеется справка
<code>CHECKPOINT</code>	Поддержка контрольных точек (RFC 1845)
<code>SIZE [<sp> <size-param>]</code>	Учет размеров писем (SMTP-сервер может оповестить о максимальном поддерживаемом размере письма: в байтах, 0 -- нет ограничения) (RFC 1870)
<code>ETRN</code>	Поддержка запросов обработки очередей писем на SMTP-сервере (RFC 1985)
<code>ENHANCEDSTATUSCODES</code>	Возвращение SMTP-сервером расширенных кодов состояния (RFC 2034)
<code>ATRN</code>	On-Demand Mail Relay SMTP с динамической IP-адресацией (RFC 2645)
<code>DELIVERBY [<min-by-time>]</code>	Доставка SMTP-сервером писем в течение указанных SMTP-клиентом интервалов времени (в секундах, сервер может оповестить о минимальном интервале, в течение которого он способен доставить письмо) (RFC 2852)
<code>PIPELINING</code>	Конвейеризация команд (RFC 2920)
<code>CHUNKING</code>	Пересылка больших писем (RFC 3030)

SMTP-расширения

10.3.5.1b

BINARYMIME	Пересылка бинарных MIME-писем (RFC 3030)
STARTTLS	TLS (RFC 3207)
DSN	Уведомления о состоянии доставки (RFC 3461)
NO-SOLICITING [<SP> <Solicitation-keywords>]	Прием SMTP-сервером писем, по умолчанию подпадающих под запрет, по просьбе (просьба может быть отклонена, сервер может оповестить о запрещенных классах писем, считающихся спамом) (RFC 3865)
MTRK	Отслеживание писем (RFC 3885)
CONPERM	Разрешение преобразования писем (RFC 4141)
CONNNEG	Согласование MIME-типов (RFC 4141)
SUBMITTER	Указание SMTP-клиентом отправителей, ответственных за письма (RFC 4405)
BURL [<SP> (imap / (imap "://" <authority>))]	Импорт писем напрямую из IMAP-сервера (RFC 4468)
FUTURERELEASE <SP> <max-future-release-interval> <SP> <max-future-release-date-time>	Указание SMTP-клиентом задержек перед доставкой писем или времен в будущем, когда нужно доставлять письма (SMTP-сервер должен оповестить о максимальном интервале времени, в течение которого он может удерживать письмо, и о дате и времени, до которых он может удерживать письмо) (RFC 4865)
AUTH <SP> <supported SASL mechanisms>	Аутентификация (в соответствии с доступными механизмами) (RFC 4954)
8BITMIME	8-мибитный MIME-транспорт (RFC 6152)
SMTPUTF8	Интернационализированная электронная почта (RFC 6531)
MT-PRIORITY [<SP> <priority-profile>]	Приоритетная обработка писем (доступны целочисленные приоритеты от -9 до 9 включая ноль) (RFC 6710)
RRVS	Учет последних достоверно известных дат и времен использования определенных почтовых ящиков определенными получателями (для обеспечения гарантии доставки писем известным владельцам почтовых ящиков если нет уверенности, что почтовые ящики по-прежнему принадлежат им) (RFC 7293)

SMTP-расширения

10.3.6.1

```
C<-S: 220 foo.com Simple Mail Transfer Service Ready
C->S: EHLO bar.com
C<-S: 250-foo.com greets bar.com
      250-8BITMIME
      250-SIZE
      250-DSN
      250 HELP
C->S: MAIL FROM:<Smith@bar.com>
C<-S: 250 OK
C->S: RCPT TO:<Jones@foo.com>
C<-S: 250 OK
C->S: RCPT TO:<Green@foo.com>
C<-S: 550 No such user here
C->S: RCPT TO:<Brown@foo.com>
C<-S: 250 OK
C->S: DATA
C<-S: 354 Start mail input; end with <CRLF>.<CRLF>
C->S: Blah blah blah...
      ...etc. etc. etc.
      .
C<-S: 250 OK
C->S: QUIT
C<-S: 221 foo.com Service closing transmission channel
```

Пример передачи электронного письма

10.3.7.1

Протокол POP (Post Office Protocol) **используют** для приема электронной почты из почтового ящика -- от почтового сервера к пользовательской станции (получателя).

Устоялась третья версия (POP3) (основное RFC -- RFC 1939).

Задействуется одно соединение.

Стандартный номер программного порта POP-сервера -- 110.

POP-сервер располагается на удаленном от пользователя почтовом сервере и имеет доступ к почтовому ящику получателя.

10.3.7.2

Любой POP-сессия последовательно проходит через три состояния:

1. AUTHORIZATION -- авторизация POP-клиента на POP-сервере.
2. TRANSACTION -- выполняется одна либо несколько транзакций с почтовым ящиком.
3. UPDATE -- взаимодействие POP-клиента с POP-сервером завершается (выделенные ресурсы освобождаются, соединение закрывается).

10.3.7.3

POP осуществляет нумерацию сообщений.

10.3.7.4

Механизм SASL предоставляет альтернативный способ POP-авторизации (в отношении POP традиционно **используют** термин «авторизация») и может сосуществовать с механизмом TLS.

10.3.8.1

Регистр букв в ROP-командах не учитывается.

10.3.8.2а

POP-команда	Описание
USER <SP> <name> <CRLF>	Имя пользователя
PASS <SP> <string> <CRLF>	Пароль
QUIT <CRLF>	Выход из удаленной системы (после этой команды POP-сервер переходит в состояние UPDATE)
STAT <CRLF>	Предоставить информацию о количестве писем в почтовом ящике и их суммарный размер (ответ должен быть в формате: +OK <SP> <nn> <SP> <mm>, где <nn> – количество, <mm> – размер в байтах; кроме писем, помеченных для удаления)
LIST [<SP> <msg>] <CRLF>	Предоставить размеры писем в почтовом ящике (построчно; ответ должен быть в формате: +OK <text> <CRLF> *(<nn> <SP> <mm> <CRLF>), где <nn> – номер, <mm> – размер; в команде может быть указан номер интересующего письма)
RETR <SP> <msg> <CRLF>	Предоставить письмо (непомеченное для удаления; содержимое письма пересыпается между ответом и строкой-точкой -- <CRLF> ". " <CRLF>)
DELE <SP> <msg> <CRLF>	Пометить сообщение для удаления (собственно перманентное удаление может быть выполнено только в состоянии UPDATE)
NOOP <CRLF>	Холостая POP-команда
RSET <CRLF>	Снять метку об удалении со всех помеченных для удаления писем
APOP <SP> <name> <SP> <digest> <CRLF>	Авторизовать с помощью имени пользователя и хэша пароля (альтернатива паре USER и PASS)
TOP <SP> <msg> <SP> <n> <CRLF>	Предоставить заголовок и n первых строк письма (при пересылке заголовок отделяется от основного текста пустой строкой)
UIDL [<SP> <msg>] <CRLF>	Предоставить уникальные идентификаторы (unique-ids listing) писем в почтовом ящике (может быть указан номер интересующего письма, поочередно)

POP-команды

10.3.8.2b

CAPA <CRLF>	Предоставить список всех возможностей, включая расширения (построчно) (RFC 2449)
STLS <CRLF>	Запустить TLS (затем должно следовать согласование TLS-параметров) (RFC 2595)
AUTH <SP> <mechanism> [<SP> <initial-response>] <CRLF>	Авторизовать в соответствии с указанным механизмом SASL (затем должна следовать авторизация; еще одна альтернатива паре USER и PASS; если используется TLS, то должна быть позже STLS) (RFC 5034)
UTF8 <CRLF>	Включить режим UTF-8 (RFC 6856)
LANG [<SP> <basic language range>] <CRLF>	Установить язык текстовых комментариев в POP-ответах (RFC 6856)

POP-команды

10.3.9.1

Предусмотрены только два РОР-ответа.

10.3.9.2

POP-ответ	Описание
+OK [<SP> "[" <resp-code> "]"] [<SP> <text>] <CRLF>	Положительный
-ERR [<SP> "[" <resp-code> "]"] [<SP> <text>] <CRLF>	Отрицательный

POP-ответы

10.3.9.3

Регистр +OK **и** -ERR должен быть именно таким.

POP-ответы могут состоять из нескольких строк.

Текстовое наполнение некоторых POP-ответов (например, на команды STAT и LIST) регламентировано.

10.3.10.1а

Согласно RFC 2449, и более новым, **выделяют** следующие возможности протокола POP (перечисляются в POP-ответе на POP-команду CAPA).

10.3.10.1b

POP-возможность	Описание
<code>TOP</code>	Поддерживается POP-команда <code>TOP</code>
<code>USER</code>	Поддерживаются POP-команды <code>USER</code> и <code>PASS</code>
<code>SASL <SP> <supported SASL mechanisms></code>	Поддерживается SASL, включая POP-команду <code>AUTH</code>
<code>RESP-CODES</code>	Поддерживаются расширенные коды в POP-ответах
<code>LOGIN-DELAY <SP> <minimum seconds between logins> [<SP> USER]</code>	Задержка перед разрешением повторной авторизации (в секундах; флаг <code>USER</code> оповещает, что значение может быть скорректировано и что значение можно запросить после авторизации)
<code>PIPELINING</code>	Поддерживается конвейеризация команд
<code>EXPIRE <SP> (<server-guaranteed minimum retention days> / NEVER) [<SP> USER]</code>	Длительность хранения писем на POP-сервере (в днях)
<code>UIDL</code>	Поддерживается POP-команда <code>UIDL</code>
<code>IMPLEMENTATION <SP> <string giving server implementation information></code>	Информационный баннер о POP-сервере
<code>STLS</code>	TLS (RFC 2595)
<code>AUTH-RESP-CODE</code>	POP-ответы при ошибках авторизации будут содержать расширенный код <code>AUTH</code> (RFC 3206)
<code>UTF8 [USER]</code>	Поддерживается кодировка UTF-8 (флаг <code>USER</code> оповещает, что эта кодировка применима и к именам пользователей и паролям) (RFC 6856)
<code>LANG</code>	Выбор языка текстовых комментариев в POP-ответах (RFC 6856)

POP-возможности

10.3.10.2

Таким образом, многие возможности (но не все) выражаются в наборах поддерживаемых POP-команд.

Некоторые возможности **конфигурируют** с помощью дополнительных аргументов.

10.3.10.3

POP-ответы на некоторые POP-команды могут содержать расширенные коды.

Допустима иерархия с разделением уровней слешем.

Регламентировано всего пять расширенных кодов ответов.

10.3.10.4

Код	Описание
LOGIN-DELAY	Авторизация не прошла успешно, так как задержка перед разрешением повторной авторизации еще не истекла (RFC 2449)
IN-USE	Авторизация прошла успешно, но почтовый ящик уже используется (например, через соединение с еще одной пользовательской станцией) (RFC 2449)
SYS/TEMP	Сбой временный и произошел по причине системной ошибки на POP-сервере (RFC 3206)
SYS/PERM	Сбой permanentный и произошел по причине системной ошибки на POP-сервере (RFC 3206)
AUTH	Ошибка связана с учетными записями пользователей (RFC 3206)
UTF8	Ошибка произошла так как письмо с содержимым в кодировке UTF-8 запрошено не в режиме UTF-8 (RFC 6856)

Расширенные коды в POP-ответах

10.3.11.1

```
C<-S: +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>
C->S: APOP mrose c4c9334bac560ecc979e58001b3e22fb
C<-S: +OK mrose's maildrop has 2 messages (320 octets)
C->S: STAT
C<-S: +OK 2 320
C->S: LIST
C<-S: +OK 2 messages (320 octets)
      1 120
      2 200
      .
C->S: RETR 1
C<-S: +OK 120 octets
      <the POP3 server sends message 1>
      .
C->S: DELE 2
C<-S: +OK message 2 deleted
C->S: QUIT
C<-S: +OK dewey POP3 server signing off
```

Пример работы с электронными письмами

10.3.12.1

Протокол IMAP (Internet Message Access Protocol) так же предназначен для приема электронной почты, но, в сравнении с POP, предоставляет комплексный функционал работы с почтовым ящиком.

Устоялась первая ревизия четвертой версии (IMAP4rev1) (основное RFC - RFC 3501).

Задействуется одно соединение.

Стандартный номер программного порта IMAP-сервера -- 143.

IMAP-сервер, так же как и POP-сервер, располагается на почтовом сервере и имеет доступ к почтовому ящику получателя.

10.3.12.2

Любой IMAP-сеанс последовательно проходит через четыре состояния:

1. Not authenticated -- IMAP-клиент еще не аутентифицирован на IMAP-сервере.
2. Authenticated -- IMAP-клиент аутентифицирован на IMAP-сервере.
3. Selected -- IMAP-клиент выбрал почтовый ящик и работает с ним.
4. Logout -- IMAP-клиент завершает взаимодействие с IMAP-сервером.

Кроме того, почтовые ящики могут быть активными (active, subscribed) и неактивными.

10.3.12.3

Бо'льшая, в сравнении с другими протоколами электронной почты, сложность IMAP требует большего разнообразия форматов данных.

Данные, передаваемые по IMAP, представляются в следующих форматах:

1. Atom -- атом (неделим) -- один либо несколько символов, не являющихся метасимволами.
2. Number -- число -- одна либо несколько цифр.
3. String -- строка -- две формы: строка в простых двойных кавычках (может быть пустой, основная форма строки) и литеральная строка (в начале число в фигурных скобках -- количество литералов, далее <CRLF>, далее сами литералы).
4. Parenthesized list -- список в круглых скобках -- элементы списка разделены <SP> (может быть пустым и вложенным).
5. NIL -- отсутствие данных (строки, списка в круглых скобках) -- метасимвол.

10.3.12.4а

Каждому письму в почтовом ящике на IMAP-сервере присваиваются следующие атрибуты:

1. Message numbers -- номера:

-- UID (Unique Identifier) -- уникальный идентификатор (32-ухбитный; уникальный в границах почтового ящика; дополняется 32-ухбитным кодом UIDVALIDITY, позволяющим проверить валидность этого идентификатора);

-- message sequence number -- последовательный номер в почтовом ящике (нумерация начинается с единицы).

2. Flags -- флаги (начинаются с метасимвола \, некоторые существуют перманентно, некоторые существуют только в течение сеанса с почтовым ящиком):

-- \Seen -- прочитано;

-- \Answered -- был дан ответ;

-- \Flagged -- срочное;

-- \Deleted -- помечено для удаления;

-- \Draft -- черновик;

-- \Recent -- новое.

3. Internal date -- дата получения (и время).

10.3.12.4b

4. Size message -- размер (в байтах).
5. Envelope structure -- структура заголовка письма (не SMTP-конверт, используемый при пересылке).
6. Body structure -- структура основного текста письма.

10.3.12.5

IMAP имеет ряд стандартизованных расширений (около сорока) и постоянно обновляется.

Так, например, расширение CONDSTORE (RFC 7162) предусматривает присвоение письму еще одного атрибута: mod-sequence -- последовательный номер модификации (64-битный, позволяет отслеживать модификации письма, формируется на основе системного времени).

Предусмотрен автоматический вывод IMAP-клиента из системы при неактивности -- отслеживается Autologout Timer (минимум полчаса).

10.3.13.1

Важной особенностью IMAP является заложенная изначально конвейеризация (исключая редкие случаи с неоднозначной интерпретацией).

Для обеспечения такого взаимодействия IMAP-клиент перед каждой IMAP-командой вставляет уникальный буквенно-цифровой тег (например, A0001). Отвечая на текущую IMAP-команду, IMAP-сервер вставляет перед IMAP-ответом такой же тег. Это позволяет IMAP-клиенту правильно определять какой IMAP-ответ к какой IMAP-команде относится.

Некоторые команды позволяют работать сразу с набором писем (sequence set). Например, 2,4:6,9:* -- ссылка на письма с последовательными номерами 2, 4, 5, 6, 9, 10 в ящике с десятью письмами.

Регистр букв в IMAP-командах не учитывается.

10.3.13.2а

IMAP-команда	Описание
Любое состояние	
CAPABILITY <CRLF>	Предоставить список возможностей
NOOP <CRLF>	Холостая IMAP-команда
LOGOUT <CRLF>	Выход из удаленной системы
ENABLE <SP> <capability names> <CRLF>	Активировать указанные расширения (RFC 5161)
Состояние not authenticated	
STARTTLS <CRLF>	Запустить TLS (затем должно следовать согласование TLS-параметров)
AUTHENTICATE <SP> <authentication mechanism name> <CRLF>	Аутентифицировать в соответствии с указанным механизмом (затем должна следовать аутентификация)
LOGIN <SP> <user name> <SP> <password> <CRLF>	Вход в удаленную систему (простая альтернатива AUTHENTICATE)

IMAP-команды

10.3.13.2b

Состояние authenticated	
<code>SELECT <SP> <mailbox name> [<SP> <parenthesized list of attribute/value pairs>] <CRLF></code>	Выбрать почтовый ящик и предоставить информацию о нем (+RFC 4466, +RFC 7162)
<code>EXAMINE <SP> <mailbox name> [<SP> <parenthesized list of attribute/value pairs>] <CRLF></code>	Предоставить информацию о почтовом ящике (+RFC 4466, +RFC 7162)
<code>CREATE <SP> <mailbox name> [<SP> <list of CREATE parameters>] <CRLF></code>	Создать почтовый ящик (+RFC 4466)
<code>DELETE <SP> <mailbox name> <CRLF></code>	Перманентно удалить почтовый ящик
<code>RENAME <SP> <existing mailbox name> <SP> <new mailbox name> [<SP> <list of RENAME parameters>] <CRLF></code>	Переназвать почтовый ящик (+RFC 4466)
<code>SUBSCRIBE <SP> <mailbox name> <CRLF></code>	Добавить почтовый ящик в список активных почтовых ящиков IMAP- сервера
<code>UNSUBSCRIBE <SP> <mailbox name> <CRLF></code>	Удалить почтовый ящик из списка активных почтовых ящиков IMAP- сервера
<code>LIST <SP> <reference name> <SP> <mailbox name with possible wildcards> <CRLF></code>	Предоставить список доступных для IMAP-клиента почтовых ящиков начиная с указанного уровня иерархии (возможно использование шаблонов)
<code>LSUB <SP> <reference name> <SP> <mailbox name with possible wildcards> <CRLF></code>	Предоставить список доступных для IMAP-клиента активных почтовых ящиков начиная с указанного уровня иерархии
<code>STATUS <SP> <mailbox name> <SP> <status data item names> <CRLF></code>	Предоставить состояние почтового ящика
<code>APPEND <SP> <mailbox name> [<SP> <flag parenthesized list>] [<SP> <date/time string>] <SP> <message literal or message (or message part) URL> <CRLF></code>	Добавить новое письмо в указанный почтовый ящик (+RFC 4466, +RFC 4469)

IMAP-команды

10.3.13.2c

Состояние selected	
CHECK <SP> <CRLF>	Установить флаг требования проверки выбранного почтового ящика
CLOSE <SP> <CRLF>	Закрыть выбранный почтовый ящик (перманентно удалить все письма, помеченные для удаления; вернуться в состояние authenticated) (+RFC 7162)
EXPUNGE <SP> <CRLF>	Перманентно удалить все письма, помеченные для удаления, из выбранного почтового ящика
SEARCH [<SP> <result specifier>] [<SP> <charset specification>] <SP> <searching criteria (one or more)> <CRLF>	Предоставить список писем из выбранного почтового ящика, отвечающих заданным критериям поиска (определенено несколько десятков критериев) (+RFC 4466, +RFC 4731, +RFC 5032, +RFC 5182, +RFC 7162)
FETCH <SP> <sequence set> <SP> <message data item names or macro> [<SP> <fetch modifiers>] <CRLF>	Предоставить указанные данные, относящиеся к указанным письмам (текст или другие части) (+RFC 4466, +RFC 7162)
STORE <SP> <sequence set> [<SP> <store modifiers>] <SP> <message data item name> <SP> <value for message data item> <CRLF>	Обновить указанные данные, относящиеся к указанным письмам (в настоящее время допускается обновление только флагов) (+RFC 4466, +RFC 7162)
COPY <SP> <sequence set> <SP> <mailbox name> <CRLF>	Скопировать указанные письма из выбранного почтового ящика в указанный почтовый ящик (добавить в конец)
UID <SP> <command name> <SP> <command arguments> <CRLF>	Два варианта. Выполнить IMAP-команду (COPY, FETCH, STORE), но в качестве аргумента <sequence set> использовать не последовательные номера письма, а UIDs. Выполнить IMAP-команду (SEARCH), но в IMAP-ответе вместо последовательных номеров писем использовать UIDs
IDLE <CRLF>	Готовность принимать IMAP-ответы-обновления о выбранном почтовом ящике в реальном времени (RFC 2177)
ESEARCH [<source options>] [<result options>] [<charset specification>] <searching criteria (one or more)> <CRLF>	Расширенный поиск (альтернатива SEARCH) (RFC 7377)
Экспериментальные расширения	
X <atom> ... <CRLF>	Выполнить нестандартную IMAP-команду (признаком является первая буква X в названии, аргументы не регламентируются)

IMAP-команды

10.3.14.1

IMAP-клиент должен быть постоянно готов к обработке любых IMAP-ответов.

Предусмотрены три формата IMAP-ответов:

1. Status responses -- ответы о состоянии.
2. Server data -- данные от IMAP-сервера.
3. Command continuation request -- запрос следующей IMAP-команды (IMAP-сервером у IMAP-клиента).

10.3.14.2

Ответы о состоянии могут быть тегированными и нетегированными.

Данные от IMAP-сервера и запросы следующей IMAP-команды всегда нетегированные.

В запросах следующей IMAP-команды вместо тега вставляется +.

В других нетегированных ответах вместо тега вставляется *.

Ответы о состоянии могут сопровождаться текстовыми комментариями (human-readable text).

В настоящее время определены следующие коды ответов.

10.3.14.3

Код	Описание
ALERT	Пользователю следует обратить внимание на текстовый комментарий
BADCHARSET [<SP> "(" < charset > * (<SP> < charset >) ")"]	Сбой поиска по причине отсутствия поддержки набора символов
CAPABILITY *(<SP> < capability > <SP> IMAP4rev1 *(<SP> < capability >)	Список возможностей (может упредждать IMAP-команду CAPABILITY)
PARSE	В текстовом комментарии описана ошибка, возникшая в процессе разбора содержимого письма
PERMANENTFLAGS <SP> "(" [< flag- perm > *(<SP> < flag-perm >)] ")"	Список перманентных флагов (список в скобках)
READ-ONLY	Выбранный почтовый ящик доступен только для чтения
READ-WRITE	Выбранный почтовый ящик доступен для чтения и для записи
TRYCREATE	Сбой при выполнении IMAP-команды APPEND либо COPY по причине недоступности почтового ящика назначения
UIDNEXT <SP> < nz-number >	Следующий UID (рассчитанный UID для ожидаемого нового письма)
UIDVALIDITY <SP> < nz-number >	UIDVALIDITY
UNSEEN <SP> < nz-number >	Последовательный номер первого непрочитанного письма
BADURL <SP> < url-resp-text >	Сбой при выполнении IMAP-команды APPEND при обработке URL (RFC 4469)
TOOBIG	IMAP-сообщение превысило 4 гигабайта (RFC 4469)
CLOSED	Текущий почтовый ящик неявно закрыт обращением к другому почтовому ящику посредством SELECT/EXAMINE (RFC 7162)
HIGHESTMODSEQ <SP> < mod- sequence-value >	Наибольший последовательный номер модификации письма (RFC 7162)
MODIFIED <SP> < sequence-set >	Набор писем с последовательными номерами модификации большими номера, указанного в IMAP-команде STORE (модифицированных после соответствующей даты и времени) (RFC 7162)
NOMODSEQ	Последовательные номера модификации писем не поддерживаются (RFC 7162)

Коды в IMAP-ответах

10.3.14.4а

IMAP-ответ	Описание
Status (состояние)	
OK <SP> ["[" <response code> "]" <SP>] <human-readable text> <CRLF>	Положительный IMAP-ответ (например, соответствующая IMAP-команда выполнена)
NO <SP> ["[" <response code> "]" <SP>] <human-readable text> <CRLF>	Ошибка при выполнении IMAP-команды
BAD <SP> ["[" <response code> "]" <SP>] <human-readable text> <CRLF>	Ошибка протокола
PREAUTH <SP> ["[" <response code> "]" <SP>] <human-readable text> <CRLF>	Аутентификация не требуется так как уже выполнена внешними средствами (вход в систему осуществлен)
BYE <SP> ["[" <response code> "]" <SP>] <human-readable text> <CRLF>	Осуществляется выход
Server and mailbox status (состояние сервера и почтового ящика)	
CAPABILITY *(<SP> <capability>)<SP> IMAP4rev1 *(<SP> <capability>) <CRLF>	IMAP-ответ на IMAP-команду CAPABILITY (должен содержать IMAP4rev1; названия нестандартных возможностей должны начинаться с x)
LIST <SP> <name attributes> <SP> <hierarchy delimiter> <SP> <name> <CRLF>	IMAP-ответ на IMAP-команду LIST (на одну команду может быть несколько ответов)
LSUB <SP> <name attributes> <SP> <hierarchy delimiter> <SP> <name> <CRLF>	IMAP-ответ на IMAP-команду LSUB
STATUS <SP> <name> <SP> <status parenthesized list> <CRLF>	IMAP-ответ на IMAP-команду STATUS (+RFC 7162)
SEARCH <SP> <zero or more numbers> <CRLF>	IMAP-ответ на IMAP-команду SEARCH (список последовательных номеров писем) (+RFC 7162)
FLAGS <SP> <flag parenthesized list> <CRLF>	Флаги (IMAP-ответ на IMAP-команду SELECT либо EXAMINE; какие флаги установлены применительно к письмам в почтовом ящике)
ESEARCH <SP> <one or more search-return-data pairs> <CRLF>	Альтернативный IMAP-ответ на IMAP-команду SEARCH (RFC 4466, +RFC 4731, +RFC 7162, +RFC 7377)
ENABLED <SP> <capability listing> <CRLF>	IMAP-ответ на IMAP-команду ENABLE (RFC 5161)

IMAP-ответы

10.3.14.4b

Mailbox size (содержимое почтового ящика)	
<number> <SP> EXISTS <CRLF>	Количество писем в почтовом ящике (IMAP-ответ на IMAP-команду <code>SELECT</code> либо <code>EXAMINE</code> , также посыпается при появлении новых писем)
<number> <SP> RECENT <CRLF>	Количество писем в почтовом ящике с флагом <code>\Recent</code>
Message status (состояние письма)	
<nz-number> <SP> EXPUNGE <CRLF>	IMAP-ответ на IMAP-команду <code>EXPUNGE</code> (письмо с указанным номером перманентно удалено, при удалении письма с определенным номером бо́льшие номера писем декрементируются, количество ответов на команду соответствует количеству удаляемых писем)
<nz-number> <SP> FETCH <SP> <message data> <CRLF>	IMAP-ответ на IMAP-команду <code>FETCH</code> (также ответ на команду <code>STORE</code> , также посыпается при изменении флагов) (+RFC 7162)
VANISHED <an EARLIER tag> <list of UIDs> VANISHED <list of UIDs>	Два варианта. Альтернативный IMAP-ответ на IMAP-команду <code>EXPUNGE</code> (при использовании расширения <code>QRESYNC</code> , один ответ с UIDs вместо нескольких ответов с последовательными номерами) (RFC 7162)
Command continuation request (запрос следующей IMAP-команды)	
["+"]<SP> <human-readable text> <CRLF>	Готовность к обработке следующей IMAP-команды (также задействуется при аутентификации)

IMAP-ответы

10.3.14.5

IMAP-ответ LIST, в том числе, отображает атрибуты соответствующего объекта :

1. \Noinferiors -- у объекта нет дочерних объектов.
2. \Noselect -- данный объект не является почтовым ящиком.
3. \Marked -- данный объект является почтовым ящиком и этот почтовый ящик «интересен» (возможно в нем появились новые письма).
4. \Unmarked -- данный объект является почтовым ящиком и этот почтовый ящик не содержит новых писем.

10.3.14.6

IMAP-ответ STATUS отображает количественные значения атрибутов почтового ящика:

1. MESSAGES -- количество писем в почтовом ящике.
2. RECENT -- количество писем с флагом \Recent.
3. UIDNEXT -- следующий UID.
4. UIDVALIDITY -- UIDVALIDITY.
5. UNSEEN -- количество писем без флага \Seen.

10.3.15.1

Стандарт очерчивает три обязательные возможности IMAP (перечисляются в IMAP-ответе на IMAP-команду CAPABILITY).

IMAP-возможность	Описание
AUTH=PLAIN	Аутентификация (на основе открытого текста)
LOGINDISABLED	Запрет IMAP-команды LOGIN
STARTTLS	TLS

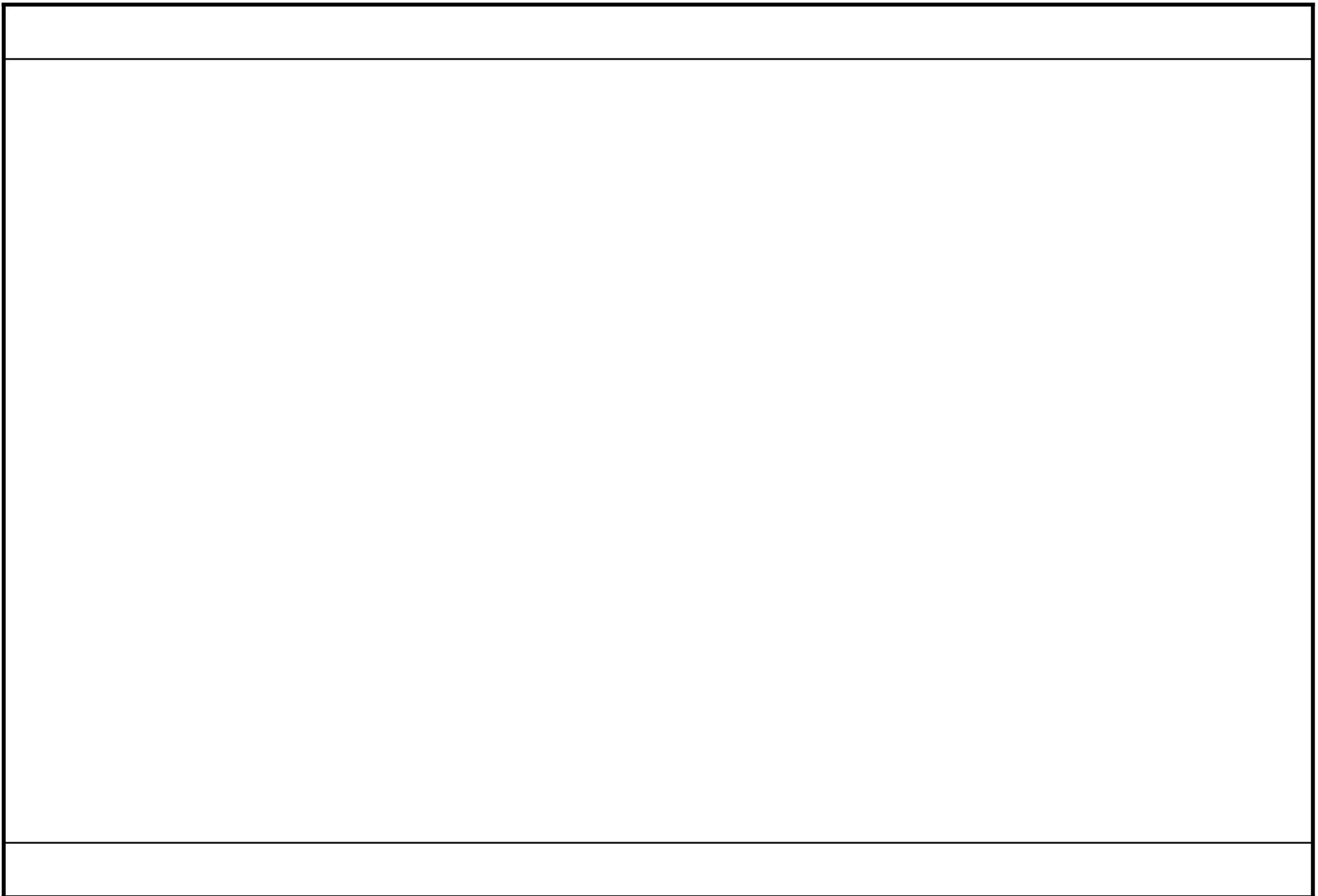
IMAP-возможности

10.3.16.1

```
C<-S: * OK IMAP4rev1 Service Ready
C->S: a001 login mrc secret
C<-S: a001 OK LOGIN completed
C->S: a002 select inbox
C<-S: * 18 EXISTS
      * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
      * 2 RECENT
      * OK [UNSEEN 17] Message 17 is the first unseen message
      * OK [UIDVALIDITY 3857529045] UIDs valid
a002 OK [READ-WRITE] SELECT completed
C->S: a003 fetch 12 full
C<-S: * 12 FETCH (FLAGS (\Seen) INTERNALDATE "17-Jul-1996 02:44:25 -0700"
  RFC822.SIZE 4286 ENVELOPE ("Wed, 17 Jul 1996 02:23:25 -0700 (PDT)"
  "IMAP4rev1 WG mtg summary and minutes"
  (("Terry Gray" NIL "gray" "cac.washington.edu"))
  (("Terry Gray" NIL "gray" "cac.washington.edu"))
  (("Terry Gray" NIL "gray" "cac.washington.edu"))
  ((NIL NIL "imap" "cac.washington.edu"))
  ((NIL NIL "minutes" "CNRI.Reston.VA.US")
  ("John Klensin" NIL "KLENSIN" "MIT.EDU")) NIL NIL
  "<B27397-0100000@cac.washington.edu>")
  BODY ("TEXT" "PLAIN" ("CHARSET" "US-ASCII") NIL NIL "7BIT" 3028
  92))
C<-S: a003 OK FETCH completed
C->S: a004 fetch 12 body[header]
C<-S: * 12 FETCH (BODY[HEADER] {342}
  Date: Wed, 17 Jul 1996 02:23:25 -0700 (PDT)
  From: Terry Gray <gray@cac.washington.edu>
  Subject: IMAP4rev1 WG mtg summary and minutes
  To: imap@cac.washington.edu
  cc: minutes@CNRI.Reston.VA.US, John Klensin <KLENSIN@MIT.EDU>
  Message-ID: <B27397-0100000@cac.washington.edu>
  MIME-Version: 1.0
  Content-Type: TEXT/PLAIN; CHARSET=US-ASCII

  )
a004 OK FETCH completed
C->S: a005 store 12 +flags \deleted
C<-S: * 12 FETCH (FLAGS (\Seen \Deleted))
  a005 OK +FLAGS completed
C->S: a006 logout
C<-S: * BYE IMAP4rev1 server terminating connection
  a006 OK LOGOUT completed
```

Пример работы с электронными письмами



10.2

Telnet

Версия 2.2

10.2.1.1

Протокол Telnet (TErminaL NETwork) (основное RFC -- RFC 854) реализует концепцию NVT (Network Virtual Terminal), уходящую корнями в UNIX-системы.

Первые UNIX-системы были «большими» и строились согласно модели сильносвязанных КС. Физические терминалы в таких системах подключались по выделенным каналам (расстояние могло быть несколько десятков метров) и поэтому их стали называть TTYs (TeleTYpes), хотя по сути они были локальными в современном понимании.

Виртуальные терминалы, то есть NVTs, представляют собой удаленные от хоста программные эмуляторы физических терминалов, связь с которыми «протянута» через СПД.

10.2.1.2

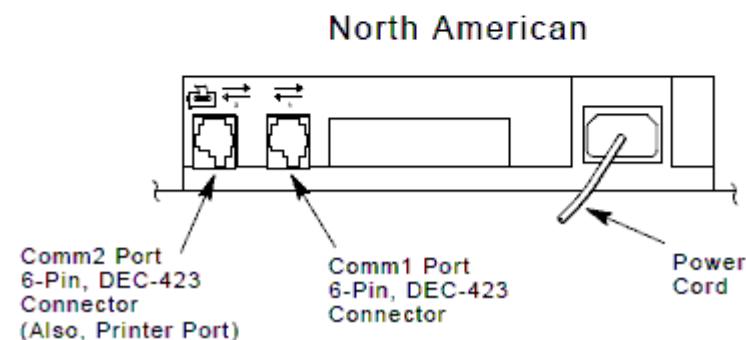
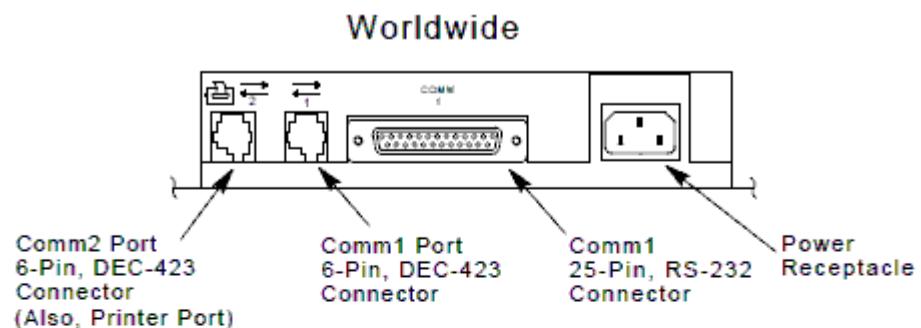
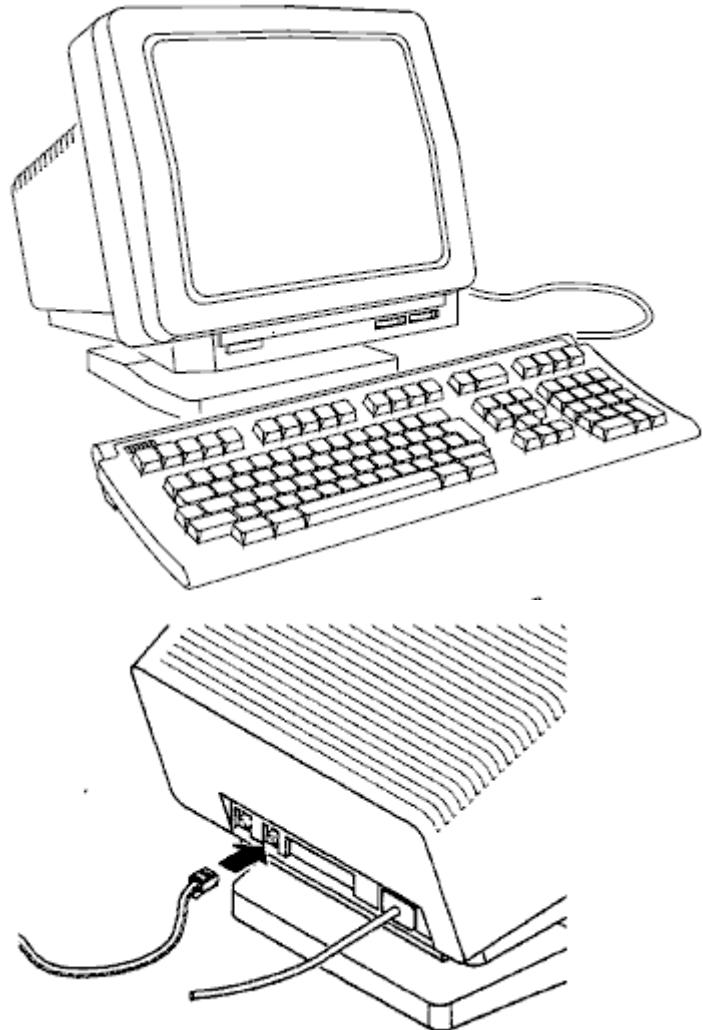
Изначально терминалы были текстовыми.

Текстовые терминалы **различают исходя из** набора поддерживаемых клавиш (основной набор, расширенный и так далее), текстового разрешения (основное -- 80x25) и некоторых других характеристик.

NVTs эмулируют основные виды текстовых терминалов.

Наиболее распространенными были следующие текстовые терминалы (выпускалось множество аналогов): DEC VT100, VT220, VT420, VT520 и IBM TN3270.

10.2.1.3



DEC VT420 [DEC]

10.2.1.4

Почти все современные терминалы -- графические.

Соответственно, графические терминалы **различают** в основном **исходя из** графического разрешения.

Удаленные графические терминалы UNIX-систем **подключают** по протоколу XDMCP (X Display Manager Control Protocol).

UNIX-программа Xterm -- стандартный эмулятор текстового терминала на графическом (на одном графическом терминале можно эмулировать сразу несколько текстовых).

10.2.1.5

Следует отметить, что хост-терминальные системы **не «вымерли»**, а **их** до сих пор достаточно широко **применяют**, особенно где беспроводные подключения не подходят по соображениям безопасности.

Усовершенствованные, значительно более компактные физические терминалы сейчас **называют** тонкими клиентами (*thin clients*).

10.2.1.6



HP t520 [HP]

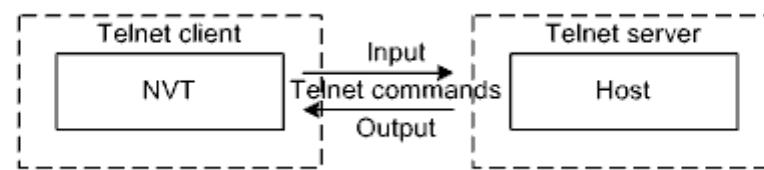
10.2.1.7

Telnet базируется на клиент-серверной модели и использует транспорт TCP.

Задействуется одно соединение.

Стандартный номер программного порта Telnet-сервера -- 23.

10.2.1.8



Структура системы Telnet

10.2.1.9

Основная задача протокола Telnet заключается в обеспечении корректной транспортировки символов потока ввода-вывода между NVT и хостом.

Используется буферизация, в том числе чтобы излишне не загружать СПД.

В режиме по умолчанию «набранные» символы отсылаются незамедлительно.

В режиме linemode (RFC 1184) символы отсылаются после нажатия Enter.

10.2.2.1

Команды самого протокола Telnet «накладываются» на основной поток ввода-вывода путем вставки в него управляющих метасимволов.

Команды могут исходить как от Telnet-клиента, так и от Telnet-сервера.

Часть команд предназначена для выполнения и на Telnet-сервере, и на Telnet-клиенте, часть -- только для выполнения на Telnet-сервере.

Некоторые команды являются запросами и поэтому зависят от команд, являющихся подтверждениями.

Признаком Telnet-команды является метасимвол <IAC> (байт со значением 255). Далее следуют код команды (один байт) и аргументы. Если необходимо переслать равный <IAC> байт, то <IAC> повторяется два раза (байт-стаффинг).

10.2.2.2

Важной частью протокола Telnet является возможность согласования параметров NVT (например, текстового разрешения).

Эти параметры выражаются в нескольких десятках Telnet-опций (под код опции так же отведен один байт).

Изучать Telnet-опции в настоящее время смысла не имеет.

10.2.2.3а

Код	Название	Описание
240	SE	Конец согласуемой Telnet-опции
241	NOP	Холостая Telnet-команда
242	Data Mark	Принудительно синхронизировать NVT с хостом с помощью экстренных TCP-данных (выполнить буферизированные Telnet-команды)
243	Break	Прервать текущий процесс (альтернатива <code>Ctrl-C</code>)
244	Interrupt Process	Прервать текущий Telnet-процесс на Telnet-сервере (завершить сеанс)
245	Abort Output	Прервать поток вывода на NVT
246	Are You There	Послать уведомление о получении данной команды («Вы на связи?»)
247	Erase Character	Удалить предыдущий символ (альтернатива <code>Backspace</code> и <code>Delete</code>)
248	Erase Line	Удалить предыдущую строку
249	Go Ahead	Ожидается следующая Telnet-команда
250	SB	Начало согласуемой Telnet-опции (далее должны следовать код опции и аргументы)
251	WILL	Два варианта. Предложение начать согласование Telnet-опции. Подтверждение начала согласования Telnet-опции. (Далее должен следовать код Telnet-опции)
252	WON'T	Два варианта. Предложение не начинать согласование Telnet-опции. Отказ начать согласование Telnet-опции. (Далее должен следовать код Telnet-опции)
253	DO	Два варианта. Начать согласование Telnet-опции. Ожидается согласование Telnet-опции. (Далее должен следовать код Telnet-опции)
254	DON'T	Два варианта. Не начинать согласование Telnet-опции. Не ожидается согласование Telnet-опции. (Далее должен следовать код Telnet-опции)
255	--	Экранированный символ 255 в потоке

Telnet-команды

10.2.3.1

Наиболее серьезным из недостатков Telnet является полная незащищенность соединения от несанкционированного доступа.

Данные, в том числе и пароли, пересылаются в виде открытого текста (plain text). В современных условиях это не может устраивать любую организацию, даже некоммерческую.

Стандарты в области защиты информации фактически запрещают применение Telnet.

Поэтому на смену Telnet пришел SSH (Secure SHell) -- идея та же, но соединение полноценно защищено.

В свое время предпринимались попытки доработать Telnet, но это направление оказалось тупиковым.

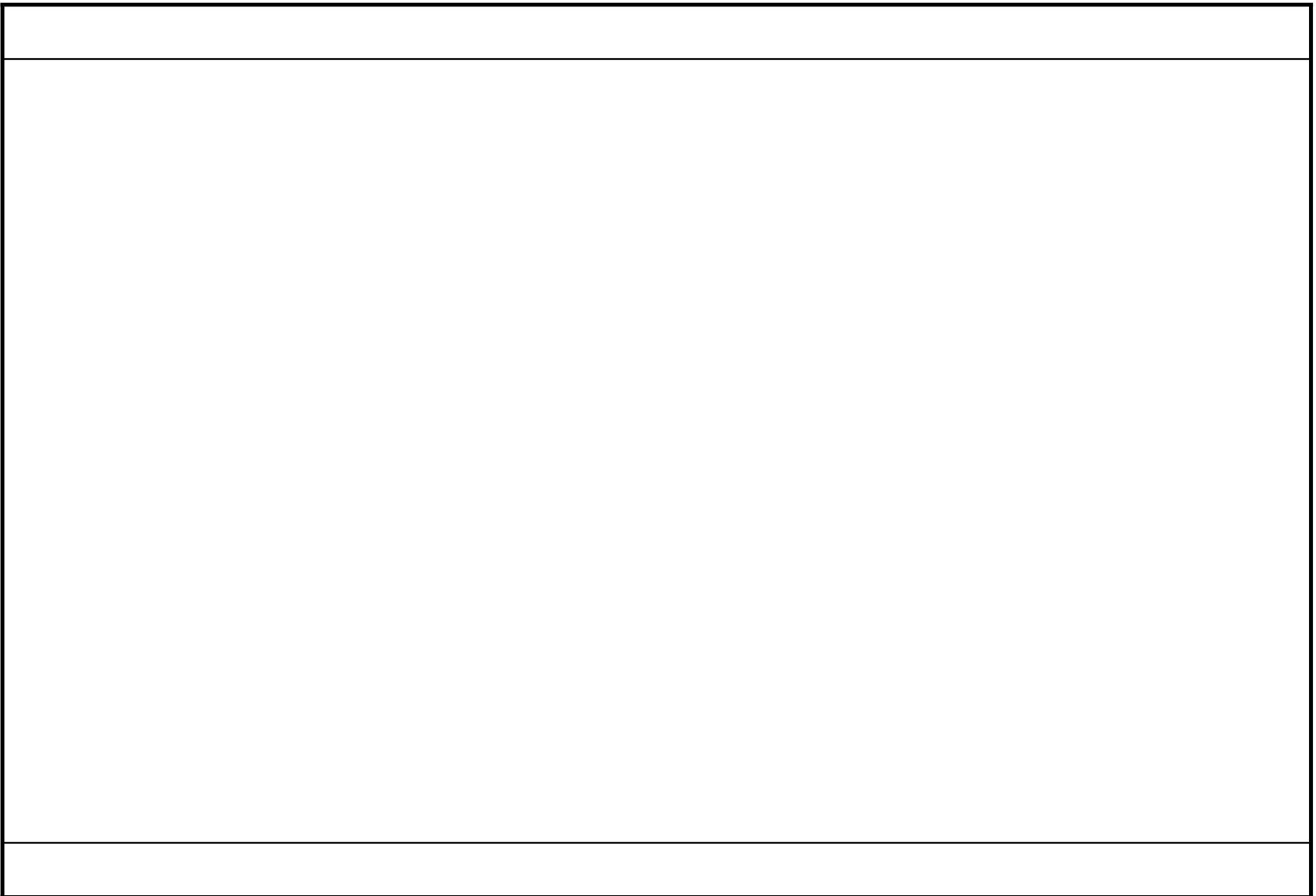
10.2.4.1

C<-S: <IAC> <DO> <NAWS>

C->S: <IAC> <WILL> <NAWS>

C->S: <IAC> <SB> <NAWS> 0 80 0 24 <IAC> <SE>

Пример согласования текстового разрешения (Telnet-опция NAWS)



СРЕДЫ ПЕРЕДАЧИ ДАННЫХ

12.0.1.1

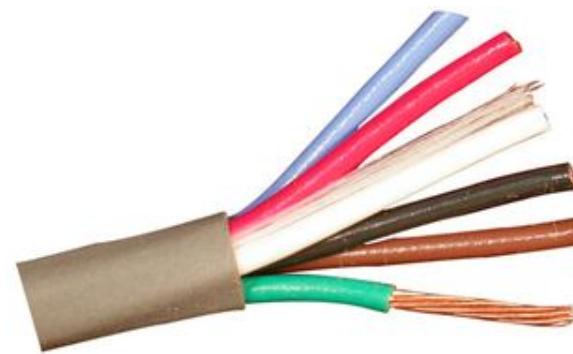
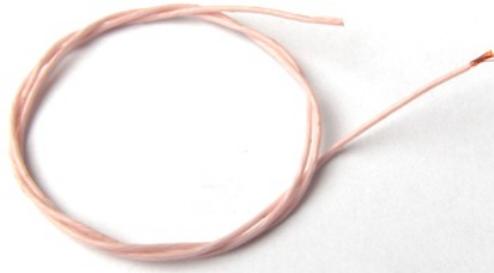
Все исконо используемые в КС СрПД можно разделить на пять типов:

1. Коаксиальные кабели (coaxials) с различным волновым сопротивлением.
2. Экранированные и неэкранированные кабели на основе витых пар (twisted pairs) различных категорий.
3. Одно- и многорежимные (одно- и многомодовые) оптоволоконные кабели (fiber равно fibre).
4. Эфир (ether).
5. Телефонные пары (phone pairs).

Где: 1, 2, 5 -- «медь» (copper); 3 -- «оптика» (optics); 1, 2, 3, 5 -- проводные (wired) СрПД; 4 -- беспроводные (wireless) СрПД.

12.0.1.2

Физически проводные СрПД выражаются в виде отдельных проводов (wires), кабелей (cables) и шлейфов (ribbon cables).



В КС в основном применяют различные кабели.

12.0.1.3

С точки зрения целевой области применения все кабели делят на:

1. *Кабели для внешней прокладки (outdoor cables)* -- СПД на улице.
2. *Кабели для внутренней прокладки (indoor cables)* -- СПД в помещениях.
3. *Оконечные кабели (cords)* -- для подключения рабочих мест.

Основные отличительные требования outdoor-кабелей: большее число проводников, высокая прочность, улучшенные электро-магнитные характеристики, влагостойкость, широкий диапазон рабочих температур, наличие дополнительных упрочняющих или гальванически развязывающих вставок.

Indoor-кабели отличаются от outdoor-кабелей меньшими габаритами и массой, большей гибкостью, лучшей пожаростойкостью, при сохранении тех же ключевых достоинств.

Кабели cords являются сравнительно простыми и низкокачественными.

12.0.1.4

В простейшем случае отдельный провод состоит из *физического проводника* (conductor) и *изоляции* (isolation).

Проводники могут быть *одножильными* (solid) и *многожильными* (stranded).

Отдельно выделяют так называемые *витые* (twisted) провода. Обычно свиты два провода, образующие дифференциальную пару.

12.0.1.5

Традиционно кабели измеряют метрами или футами ($1 \text{ ft} = 30,48 \text{ sm}$).

Сечение проводников, используемых в КС (и не только), принято измерять в AWG (American Wire Gauge): диаметр 1 mm соответствует 18 AWG (сечение $0,78 \text{ mm}^2$; максимальный ток 2,36 A -- при максимально допустимой плотности тока 3 A/mm^2).

Например, стандартное сечение жилы витой пары равно 24 AWG (диаметр около 0,5 mm).

12.0.1.6

Многие сведения о кабеле, в частности соответствие стандартам, производители указывают при его маркировке.



12.0.1.7

Любой разъем (connector) состоит из вилки (male) и розетки (female).

Контакты разъемов могут быть либо штыревыми, либо гнездовыми.

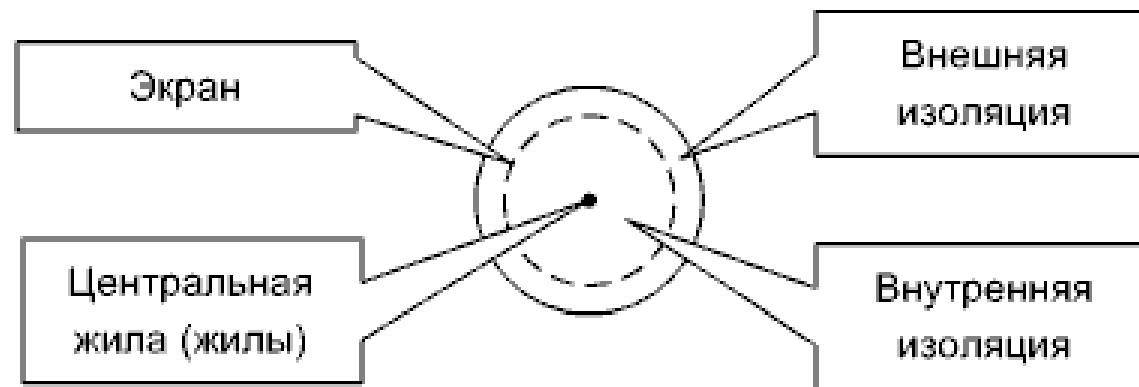
В настоящее время для соединения разъемов с проводами пайку практически не используют. Следовательно, широко применяют специальные инструменты и почти всегда отсутствуют соответствующие пайке специальные покрытия проводников.

12.0.2.1

В сегментах КС широко использовали три базовых вида коаксиальных кабелей: с волновым сопротивлением $50\ \Omega$ -- RG-8, RG-58, и с волновым сопротивлением $75\ \Omega$ -- RG-59.

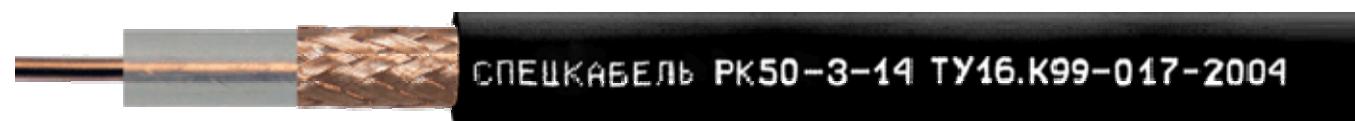
Коаксиальные outdoor- и indoor-кабели отличаются от cord-кабелей в основном внешней изоляцией.

12.0.2.2



Структура коаксиального кабеля

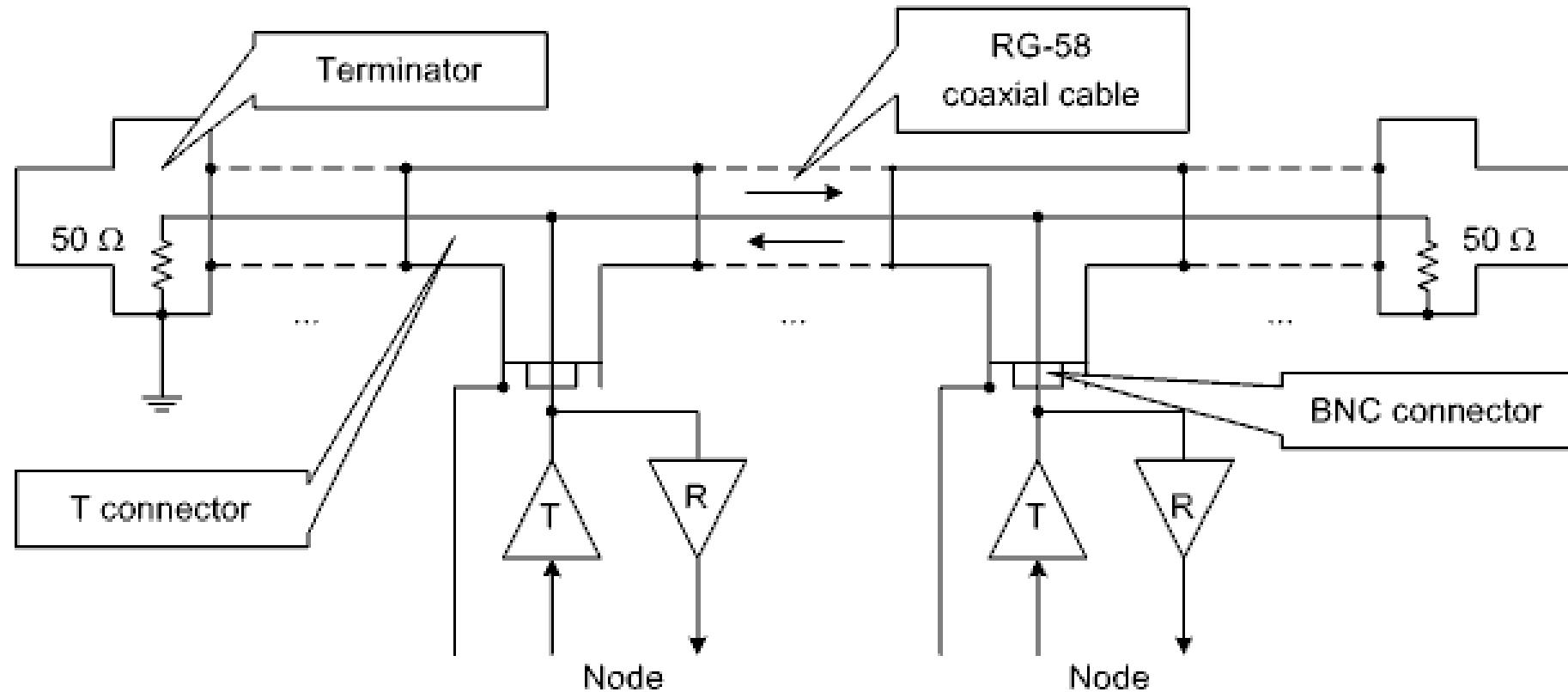
12.0.2.3



РК 50 (аналог RG-58) [Спецкабель]

12.0.2.4

Для формирования сегмента на базе коаксиального кабеля необходимо соответствующее количество BNC-разъемов (Bayonet-Neill-Concelman), Т-соединителей и пара *терминаторов* (terminators), один из которых заземляют.



Пример структуры сегмента с исп. коаксиального кабеля (10BASE2)

12.0.2.5



BNC connector; Terminator, T connector, Barrel

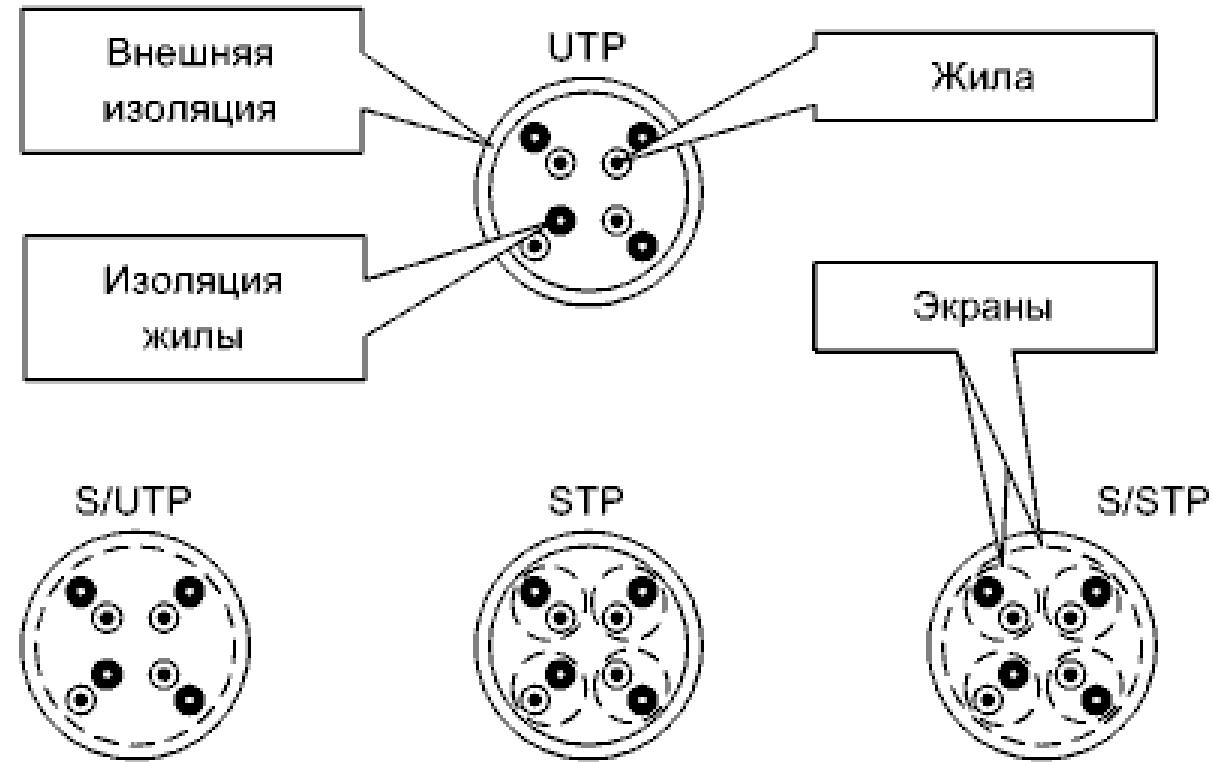
12.0.2.6

Коаксиальные кабели производители обычно выпускают черными, реже серыми.

12.0.3.1

В сегментах КС широко используют четыре основных вида кабелей на основе витых пар.

12.0.3.2

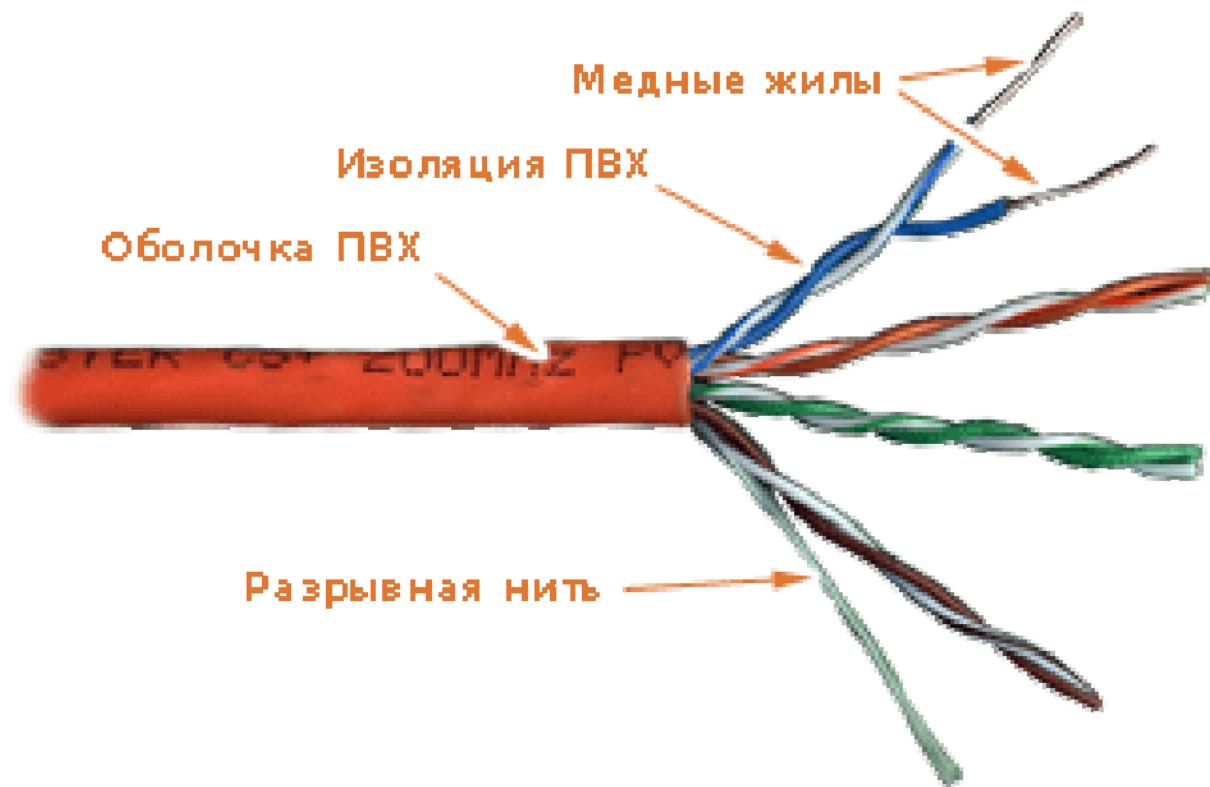


Где ТР -- Twisted Pair, S -- Shielded, U -- Unshielded, плюс может быть F -- Foiled (если для изготовления экрана применена фольга).

Особо выделяют плоский (flat) кабель (**например**, для напольной прокладки).

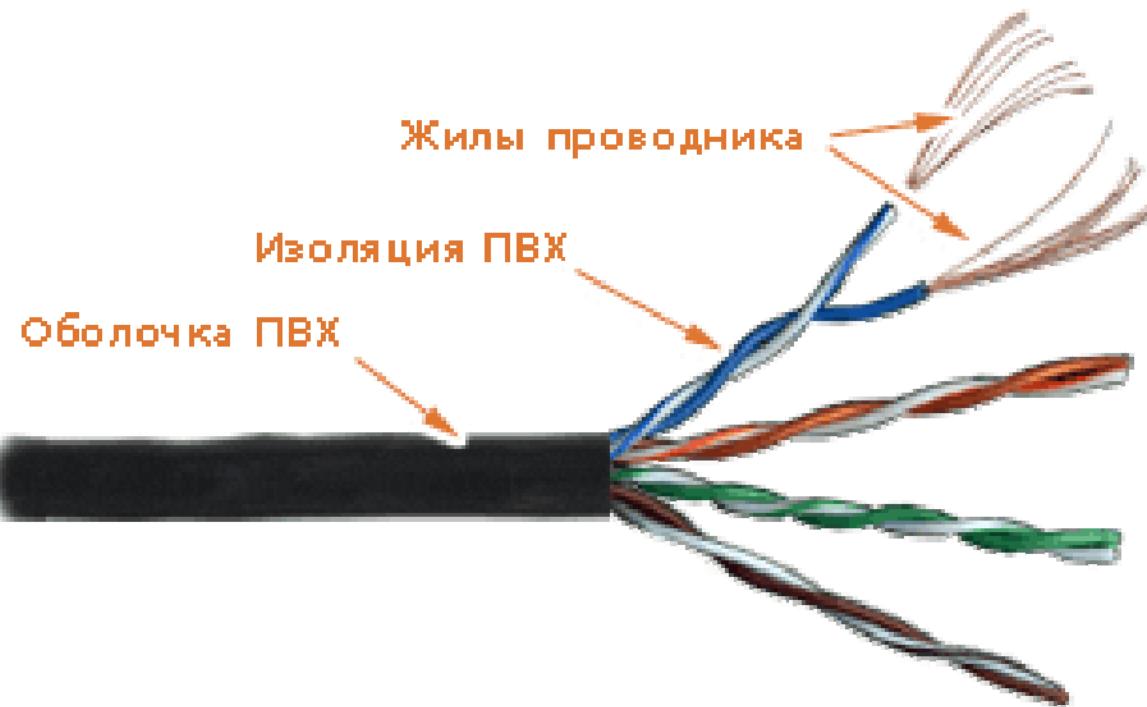
Структура кабелей на основе витых пар

12.0.3.3а



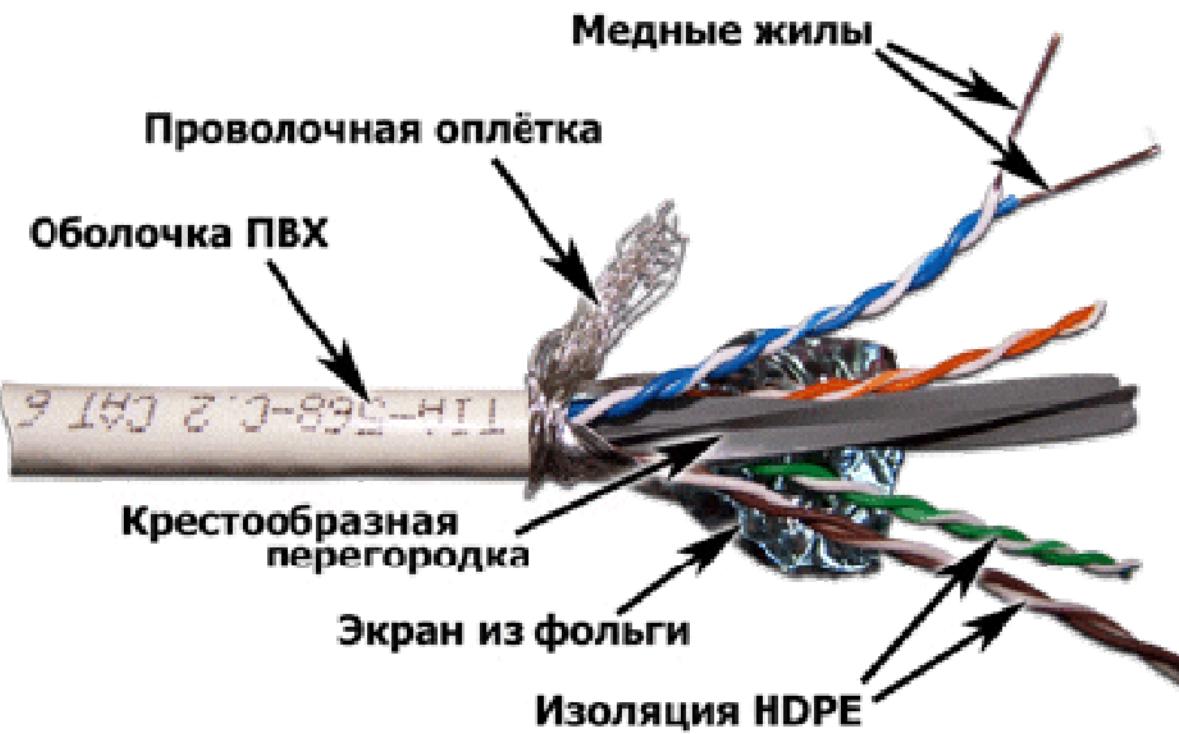
Solid UTP category 5e [Lanmaster]

12.0.3.3b



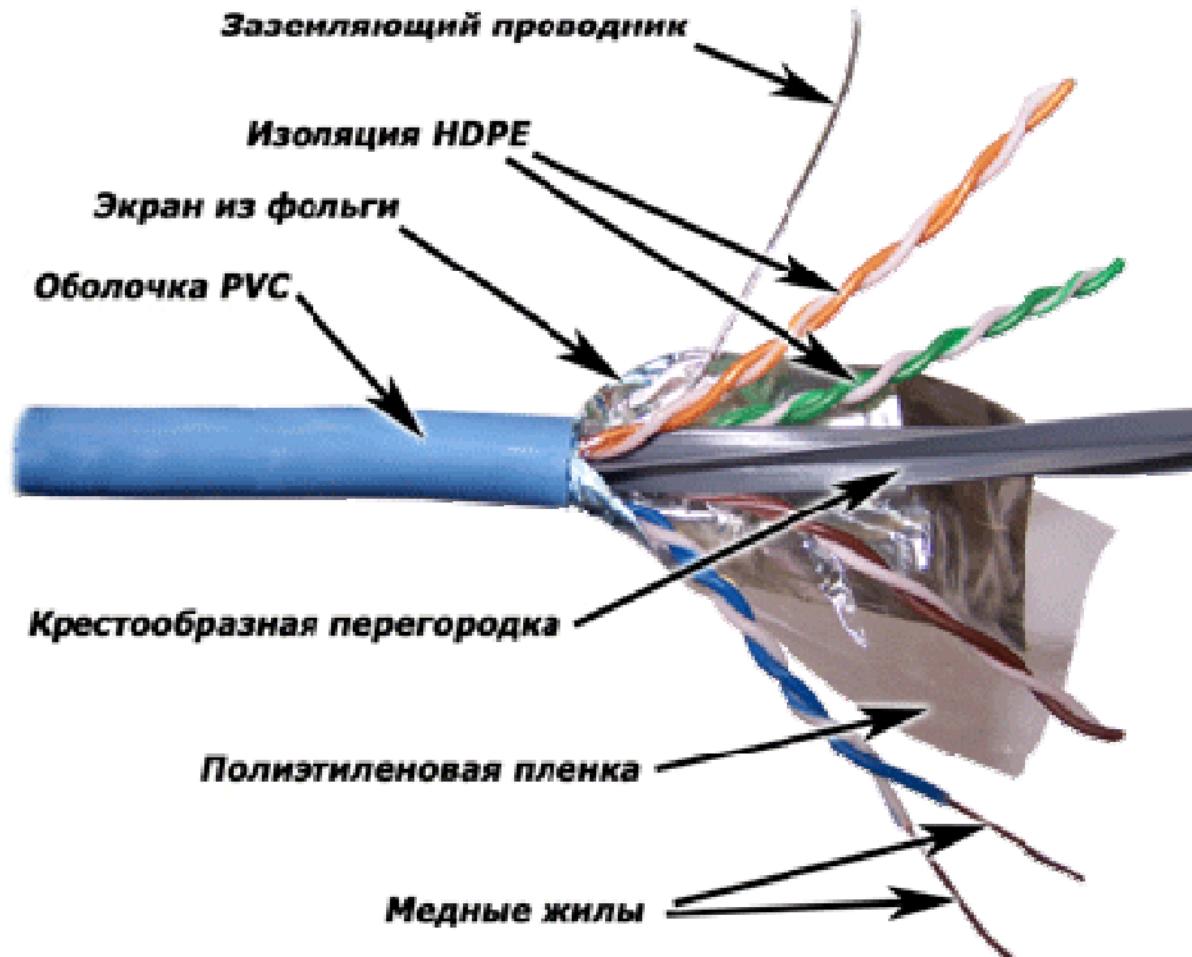
Stranded UTP category 5e [Lanmaster]

12.0.3.3c



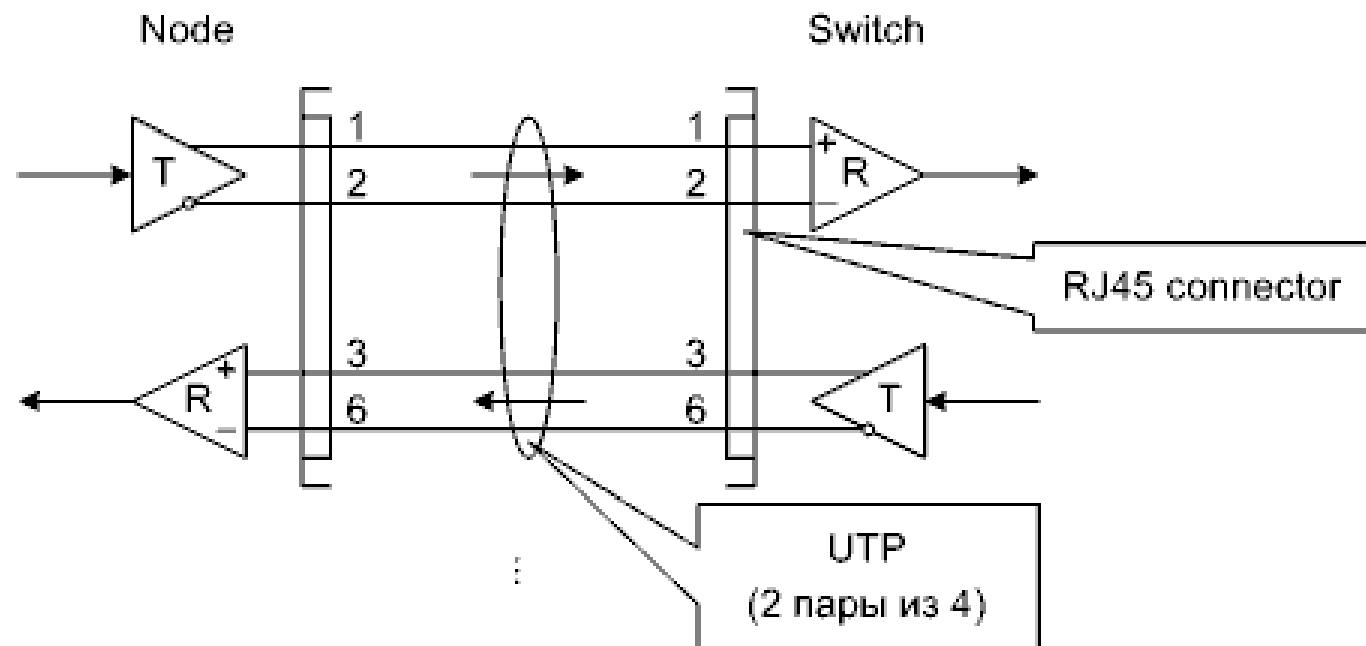
Solid SF/UTP category 6 [Lanmaster]

12.0.3.3d



Solid F/UTP category 6A [Lanmaster]

12.0.3.4



Пример структуры сегмента с использованием витых пар (100BASE-TX)

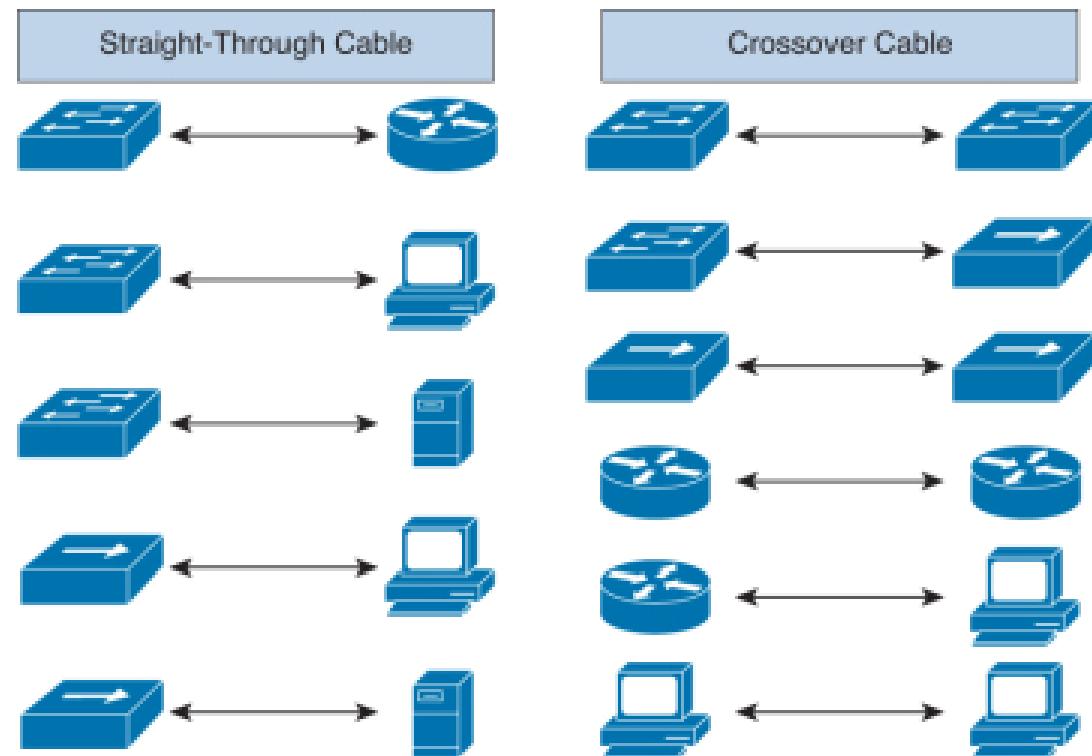
12.0.3.5

В типовых случаях, витыми парами соединяют разноранговое сетевое оборудование. Например, пользовательскую станцию подключают к коммутатору, или коммутатор подключают к маршрутизатору. При этом используют кабели с «прямой» разводкой.

При необходимости, для соединения однорангового оборудования, например непосредственного связывания двух пользовательских станций, используют кросс-кабели -- пары TD и RD скрещены.

(Полная аналогия с вариантами соединений ООД и АПД.)

12.0.3.6



Межсоединения сетевого оборудования по правилам Cisco [Cisco]

12.0.3.7

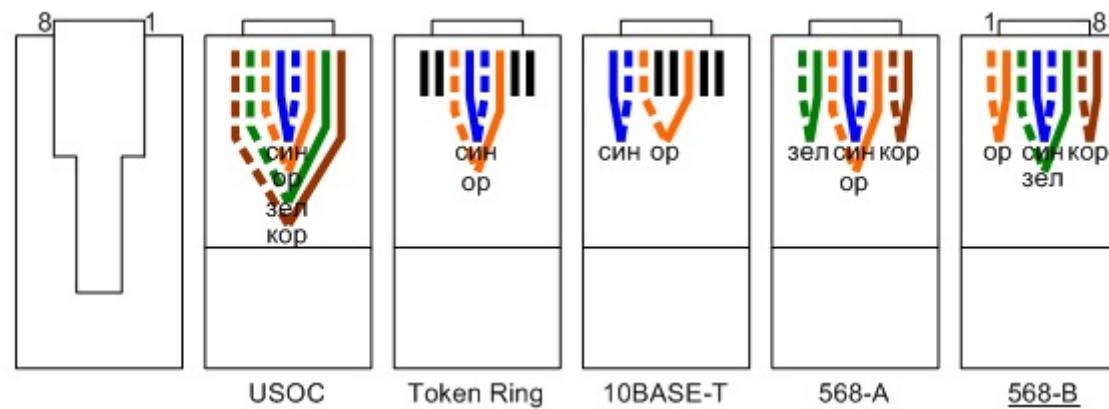
Для подключения кабелей на основе витых пар применяют разъемы RJ45.

12.0.3.8



RJ45 connector

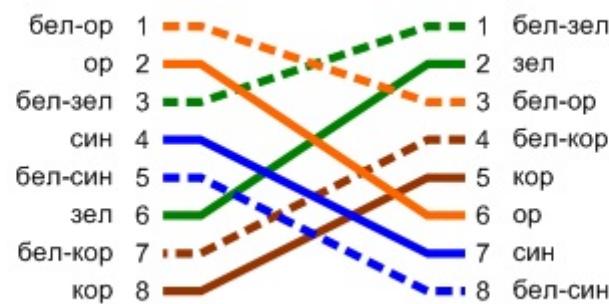
12.0.3.9



(У нас традиционно выбирают вариант 568-В.)

Стандартная разводка витых пар

12.0.3.10



Кросс-кабель Gigabit Ethernet

12.0.3.11

Цвета самих кабелей (и колпаков разъемов) в стандартах не оговорены. От производителей более-менее доступны кабели 12 стандартных цветов (привязаны к палитре RAL).

Обычные кабели имеют серый цвет (различные оттенки). Другие цвета (например, оранжевый или, даже, белый) «говорят» о более высоком качестве (например, лучшей пожарной безопасности). И востребованы для маркировки кабельных систем.

12.0.4.1

Используемые оптоволоконные кабели отличаются большим разнообразием -- следствие относительной дороговизны.

12.0.4.2

Рабочими компонентами оптоволоконных кабелей являются *световоды* (primary fiber, waveguide, lightpipe), изготовленные из оптоволокна, то есть особого кварцевого стекла. Поскольку оптоволокно очень хрупкое, его многократно защищают различными способами. **Световод -- это оптический волновод.**

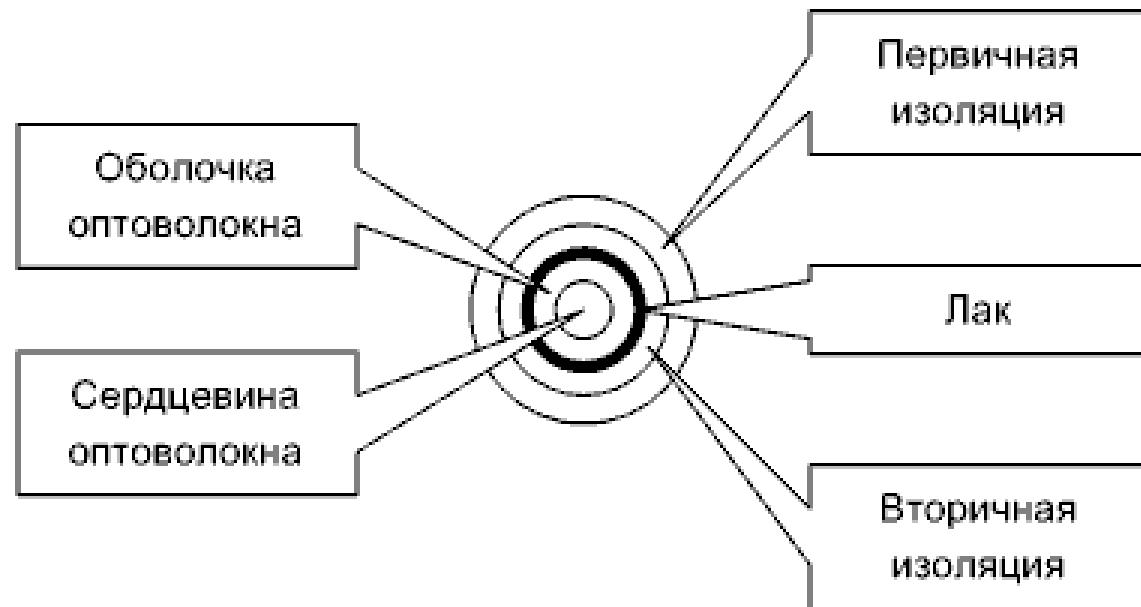
Рабочими компонентами самого световода являются *оболочка* (cladding) и *сердцевина* (core).

12.0.4.3



[Amazon]

12.0.4.4



Структура световода

12.0.4.5

В стандартах предусмотрены восемь базовых видов световодов: ОМ1, ОМ2, ОМ3, ОМ4 и ОМ5 -- многорежимные; OS1, OS2 и OS1a -- однорежимные (по-другому MM1, MM2, MM3, MM4, MM5; SM1, SM2, SM1a соответственно).

Отличаются полосой пропускания и другими техническими характеристиками.

Диаметр сердцевины: 62,5 $\mu\text{м}$ (американский стандарт) -- ОМ1; 50 $\mu\text{м}$ (европейский стандарт) -- ОМ2, ОМ3, ОМ4 и ОМ5; 9 $\mu\text{м}$ -- OS1, OS2 и OS1a.

Диаметр оболочки: 125 $\mu\text{м}$ -- для всех видов.

Общий же диаметр световода, с учетом буферизации, обычно равен около 250 $\mu\text{м}$ (может быть до 1 мм).

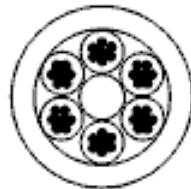
12.0.4.6

Применяют множество видов оптоволоконных кабелей.

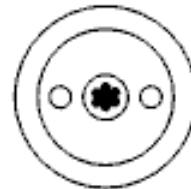
Outdoor,
Indoor:



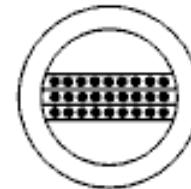
С профицированным
сердечником



Модульный

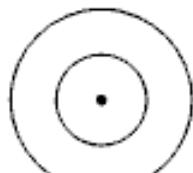


С центральной
трубкой



Ленточный

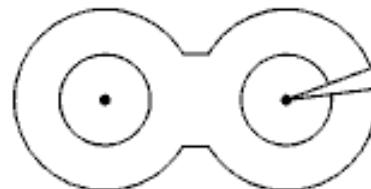
Cords:



Симплексный



Круглый
дуплексный



Zip-дуплексный

Световод

Дополнительно все оптоволоконные кабели делят на два подтипа:

1. Содержащие металлизированные упрочняющие конструкции или проводники.
2. Полностью диэлектрические.

Примеры структур оптоволоконных кабелей различного назначения

12.0.4.7a



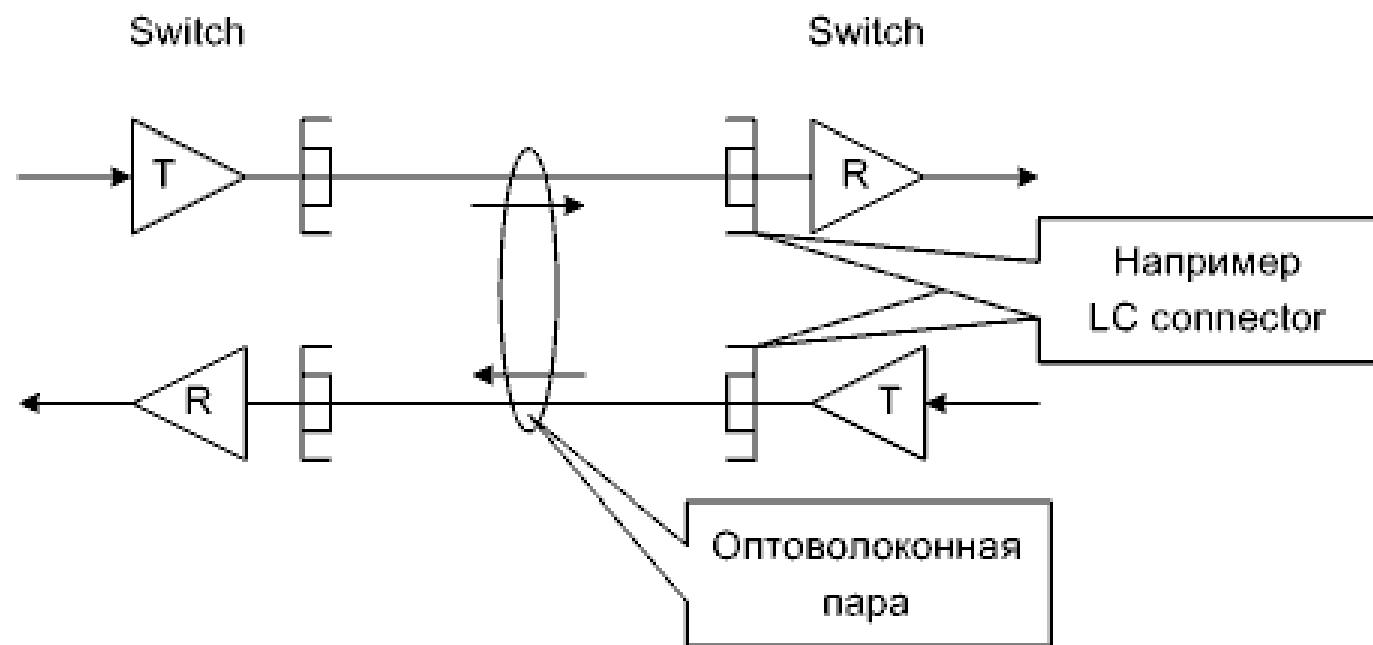
SM zip fiber [Lanmaster]

12.0.4.7b



MM modular fiber [Lanmaster]

12.0.4.8



Пример структуры сегмента с использованием оптоволокна (1000BASE-SX)

12.0.4.9

Оптоволоконные соединения выполняют двумя способами:

1. Разъемным, причем может быть:

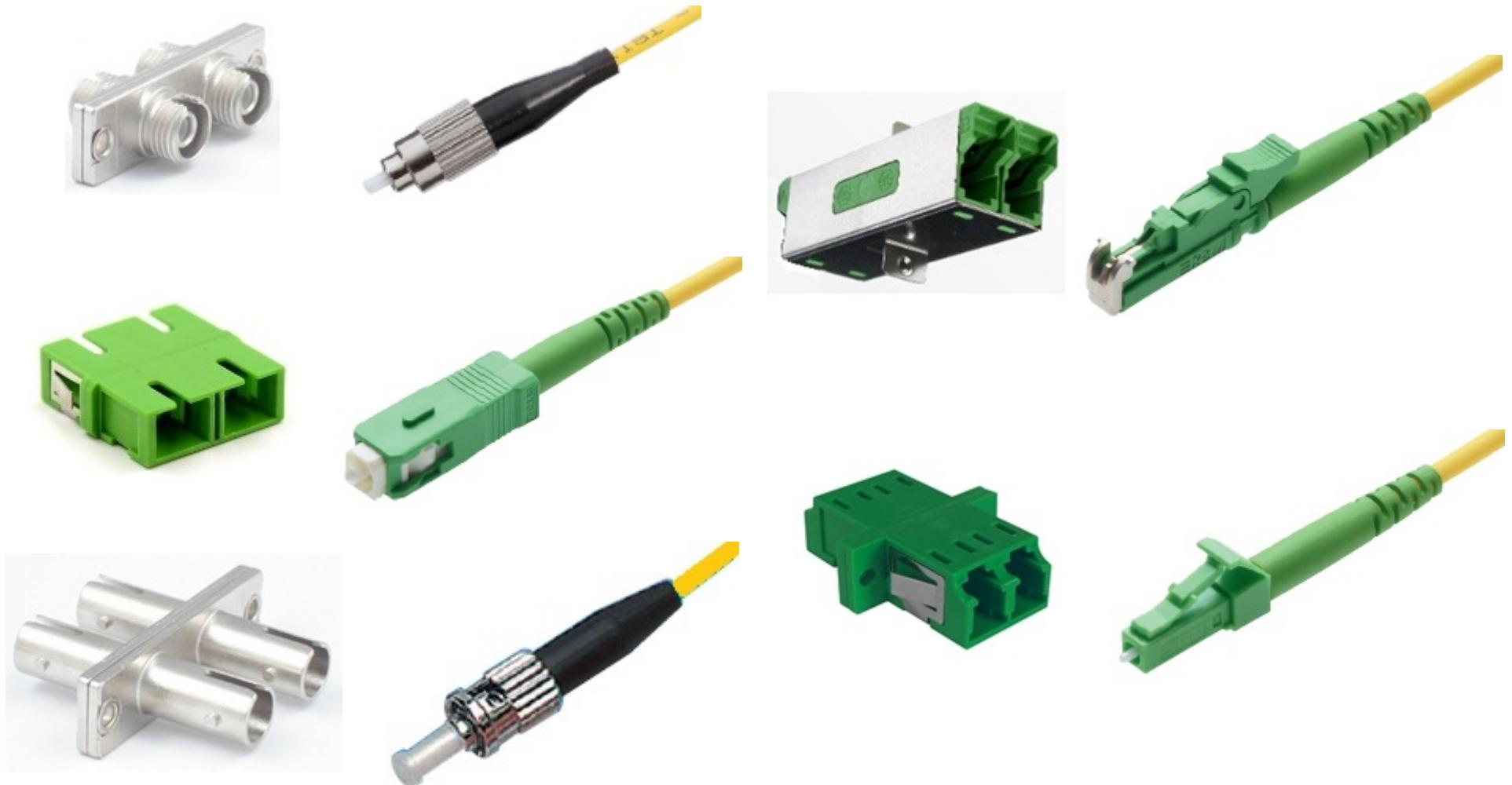
- контактным;
- линзовым.

2. Неразъемным, причем может быть:

- сплавным;
- механическим.

Оптоволоконные разъемы так же отличаются большим разнообразием. Разработано около 100 типов. Основные стандартные: FC, SC, ST (менее компактные); E-2000 (LSH), LC (более компактные).

12.0.4.10



FC, SC, ST (слева); E-2000, LC (справа) connectors [R&M, PFP, Fibertronics, Euromicron, CyberXLink]

12.0.4.11а

Еще одно отличие заключается в том, что в стандартах оговорена цветовая маркировка оптоволоконных световодов, кабелей, разъемов, а также модулей (со световодами разных видов) в составе кабелей.

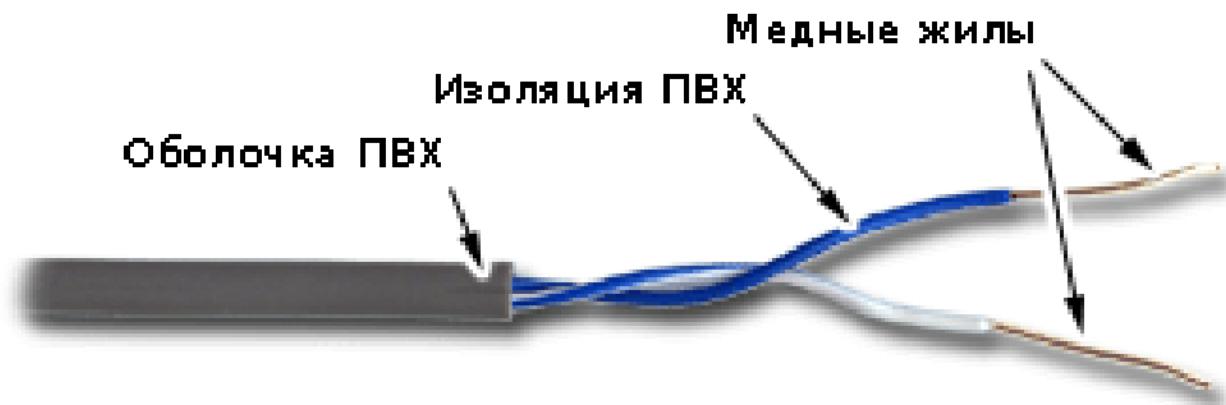
Если цветовая маркировка по тем или иным причинам не подходит, то, как альтернатива, предусмотрена маркировка штриховкой.

12.0.4.11b

Кабели cords, indoor и, по возможности, outdoor		
Вид световодов	TIA-598	IEC 60794-2 (EN 60794-2)
OM1	оранжевый (orange)	серый (grey)
OM2	оранжевый (orange)	оранжевый (orange)
OM3	аквамариновый (aqua)	бирюзовый (turquoise)
OM4	аквамариновый (aqua)	пурпурный (magenta)
OM5	лаймовый (lime)	лаймовый (lime)
OS1, OS2, OS1a	желтый (yellow)	желтый (yellow)
Световоды		
Номер	TIA-598	IEC 60794-2 (EN 60794-2)
1	светло-синий (blue)	светло-синий (blue)
2	оранжевый (orange)	желтый (yellow)
...
Разъемы		
Вид световодов	TIA-568.3	ISO/IEC 11801, EN 50173-1
OM1	бежевый (beige)	бежевый (beige) либо черный (black)
OM2	черный (black)	черный (black) либо бежевый (beige)
OM3	аквамариновый (aqua)	-- но обычно бирюзовый (turquoise)
OM4	аквамариновый (aqua)	-- но обычно фиолетовый (violet)
OM5	лаймовый (lime)	-- но обычно лаймовый (lime)
OS1, OS2, OS1a	светло-синий (blue)	светло-синий (blue)
OS1, OS2, OS1a (угловой физический контакт)	зеленый (green)	зеленый (green)

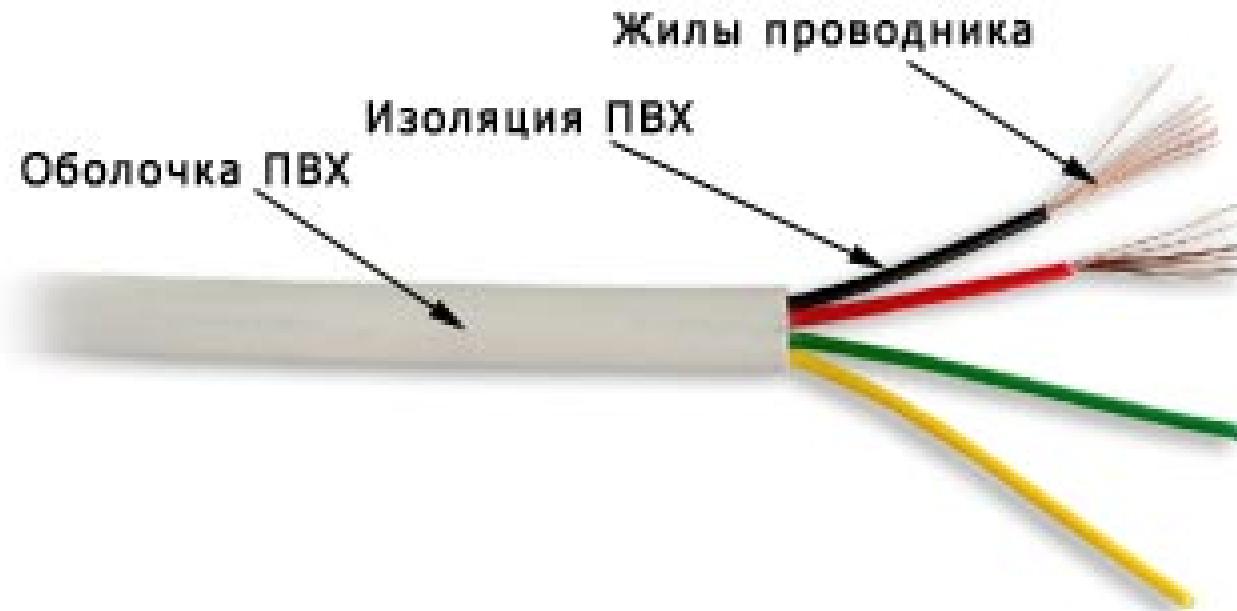
Цветовая маркировка оптоволоконных кабелей

12.0.5.1а



Phone pair [Lanmaster]

12.0.5.1b



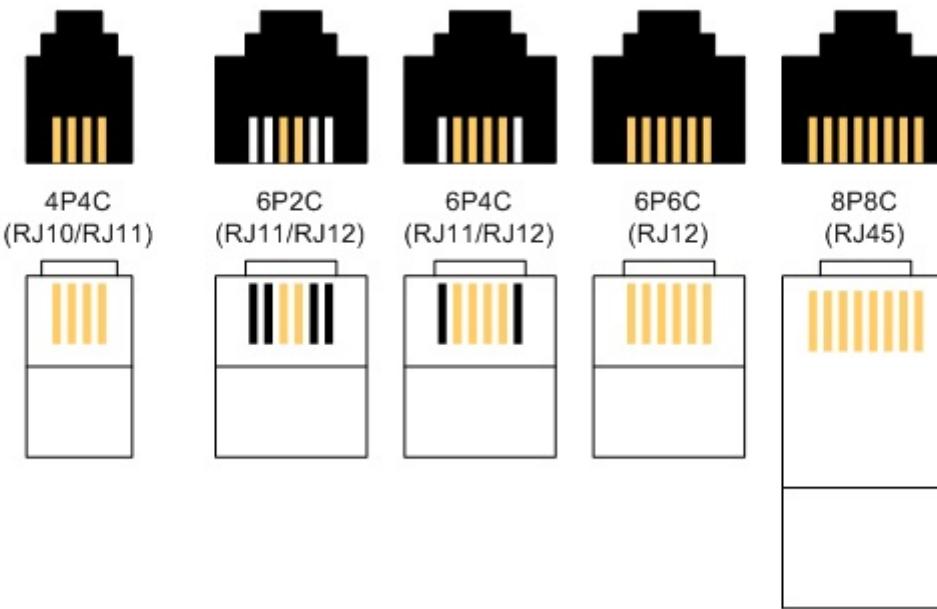
Phone pair [Lanmaster]

12.0.5.1с



Phone outdoor cable [Lanmaster]

12.0.5.2



(Названия RJ11 и RJ12 вариативны, как и некоторые другие названия из группы RJ неоднозначны, в разных «серьезных» источниках фигурируют по-разному.)

Основные телефонные разъемы типа RJ

12.0.6.1

Реализации СрПД ЛКС, как правило, соответствуют стандартам семейства IEEE 802.x.

12.0.6.2а

Коаксиальный кабель	Витая пара	Оптоволокно	Специальные среды
Ранние реализации Ethernet (около 1 Mbit/s)			
Xerox Ethernet	1BASE5 StarLAN1	—	2BASE-TL (RAS, phone pair)
Ethernet (10 Mbit/s)			
10BASE5 10BASE2	10BASE-T StarLAN10	FOIRL 10BASE-FB 10BASE-FL 10BASE-FP	10BROAD36 (RAS, coaxial) 10PASS-TS (RAS, phone pair)
Fast Ethernet (100 Mbit/s)			
—	100BASE-TX 100BASE-T2 100BASE-T4	100BASE-FX 100BASE-SX	100BASE-BX10 (RAS, fiber) 100BASE-LX10 (RAS, fiber) 100BASE-T1 (industrial, twisted pair)
Gigabit Ethernet (1 Gbit/s)			
1000BASE-CX (twinaxial)	1000BASE-T 1000BASE-TX	1000BASE-SX 1000BASE-LX 1000BASE-LX10 1000BASE-EX 1000BASE-ZX	1000BASE-BX10 (RAS, fiber) 1000BASE-PX10 (RAS, fiber) 1000BASE-PX20 (RAS, fiber) 1000BASE-KX (cluster, backplane) 1000BASE-T1 (industrial, twisted pair)
Gigabit Ethernet (Multigigabit)			
—	2.5GBASE-T (NBASE-T)	—	—
—	5GBASE-T (NBASE-T)	—	—
Gigabit Ethernet (10 Gbit/s)			
10GBASE-CX4 (4 x twinaxial) 10GBase-CR (SFP+ Direct Attach) (twinaxial)	10GBASE-T	10GBASE-SR 10GBASE-LR 10GBASE-ER 10GBASE-LX4 10GBASE-LRM	10GBASE-SW (WAN, SONET) 10GBASE-LW (WAN, SONET) 10GBASE-EW (WAN, SONET) 10GBASE-ZR (WAN, SONET/SDH) 10GBASE-KR (cluster, backplane) 10GBASE-KX4 (cluster, backplane) 10GBASE-PR (RAS, EPON) 10/1GBASE-PRX (RAS, EPON) 10GPASS-XR (RAS, coaxial)
Gigabit Ethernet (25 Gbit/s)			
25GBASE-CR (twinaxial)	25GBASE-T	25GBASE-SR	25GBASE-KR (cluster, backplane)
Gigabit Ethernet (40 Gbit/s)			
40GBASE-CR4 (4 x twinaxial)	40GBASE-T	40GBASE-SR4 40GBASE-LR4 40GBASE-ER4	40GBASE-KR4 (cluster, backplane) 40GBASE-FR (WAN, SONET/SDH)
Gigabit Ethernet (100 Gbit/s)			
100GBASE-CR10 (10 x twinaxial) 100GBASE-CR4 (4 x twinaxial)	—	100GBASE-SR10 100GBASE-LR4 100GBASE-ER4 100GBASE-SR4	100GBASE-KP4 (cluster, backplane) 100GBASE-KR4 (cluster, backplane)

Физический уровень Ethernet. (Серым цветом выделены не IEEE-стандарты)

12.0.6.2b

Где подчеркнуты ключевые использовавшиеся либо используемые стандарты:

10BASE5 (1983) -- «толстый» (thick) коаксиальный кабель 50Ω (до 500 м) плюс внешние приемопередатчики;

10BASE2 (802.3a, 1985) -- «тонкий» (thin) коаксиальный кабель 50Ω (до 185 м) плюс интегрированные приемопередатчики;

10BASE-T (802.3i, 1990) -- две телефонные витые пары (до 100 м);

10BASE-FL (802.3j, 1993) -- два многорежимных световода (до 500 м) плюс нечетко регламентированные источники излучения (обычно LEDs);

100BASE-TX (802.3u, 1995) -- две неэкранированные либо экранированные витые пары категории 5 (до 100 м);

100BASE-FX (802.3u, 1995) -- два многорежимных световода (до 2 km) (реализации поддерживают и однорежимные световоды длиной десятки километров) плюс нечетко регламентированные источники излучения (реализации поддерживают LEDs и лазеры);

12.0.6.2c

1000BASE-SX (802.3z, 1998) -- два многорежимных световода (до 275 м -- 62,5 μ m, до 550 м -- 50 μ m) плюс коротковолновые (short wavelength) лазеры (770 -- 860 nm);

1000BASE-LX (802.3z, 1998) -- два однорежимных (до 5 km) либо многорежимных световода (до 550 м) плюс длинноволновые (long wavelength) лазеры (1270 -- 1355 nm);

1000BASE-T (802.3ab, 1999) -- четыре неэкранированные либо экранированные витые пары категории 5 (до 100 m);

2.5GBASE-T (802.3bz, 2016) -- четыре неэкранированные либо экранированные витые пары категории 5e (расстояние до 100 m);

5GBASE-T (802.3bz, 2016) -- четыре неэкранированные либо экранированные витые пары категории 5e (расстояние до 100 m);

12.0.6.2d

10GBASE-SR (802.3ae, 2002) -- два многорежимных световода (до 33 м -- 62,5 им, до 400 м -- 50 им) плюс коротковолновые лазеры (840 -- 860 nm);

10GBASE-LR (802.3ae, 2002) -- два однорежимных световода (до 10 km) плюс длинноволновые лазеры (1310 nm);

10GBASE-ER (802.3ae, 2002) -- два однорежимных световода (до 30 km) плюс экстрадлинноволновые (extra long wavelength) лазеры (1550 nm);

10GBASE-T (802.3an, 2006) -- четыре неэкранированные (до 55 м) либо экранированные (до 100 м) витые пары категории 6, либо четыре неэкранированные либо экранированные витые пары категории 6A (до 100 м).



СТРУКТУРИРОВАННЫЕ КАБЕЛЬНЫЕ СИСТЕМЫ

13.0.1.1

Структурированная кабельная система (СКС) -- Structured Cabling System (SCS) здания либо сооружения -- это упорядоченная по тем или иным критериям совокупность телекоммуникационных и силовых кабелей, различного сетевого оборудования, а также соответствующих специализированных помещений.

13.0.2.1

	Международные, ISO/IEC	Американские, ANSI/TIA/EIA	Европейские, EN
Проектирование	<u>11801-1</u> (общее), <u>11801-2</u> (офисы), <u>11801-3</u> (индустрия), <u>11801-4</u> (жилые дома), <u>11801-5</u> (ЦОД), <u>11801-6</u> (интеграция услуг)	<u>568.0</u> (общее), <u>568.1</u> (коммерческие здания), <u>568.2</u> (кабели на основе витых пар), <u>568.3</u> (оптоволоконные кабели), <u>568.4</u> (специальные коаксиальные кабели), <u>570</u> (интеграция услуг), <u>942</u> (ЦОД), <u>1005</u> (индустрия)	<u>50173-1</u> (общее), <u>50173-2</u> (офисы), <u>50173-3</u> (индустрия), <u>50173-4</u> (жилые дома), <u>50173-5</u> (ЦОД), <u>50173-6</u> (интеграция услуг)
Монтаж	<u>14763-2</u> (часть об установке), <u>14763-3</u> (тестирование оптоволоконных кабелей), <u>30129</u> (заземление)	<u>569</u> (прокладка кабелей), <u>607</u> (заземление)	<u>50174-1</u> (общее), <u>50174-2</u> (indoor), <u>50174-3</u> (outdoor), <u>50310</u> (заземление), <u>50346</u> (тестирование)
Эксплуатация	<u>14763-2</u> (часть об администри- ровании)	<u>606</u> (администри- рование)	--

Основные действующие стандарты СКС

13.0.3.1

Основой для построения любой СКС является древовидная топология, узлами которой служит сетевое оборудование определенного типа (distributors).

В связи с этим, технические помещения СКС (так же distributors) делят на два типа:

1. Кроссовые (telecommunications rooms).
2. Аппаратные (equipment rooms).

Аппаратные отличаются от кроссовых тем, что в них, наряду с активным, пассивным, монтажным и вспомогательным сетевым оборудованием, может быть размещено серверное оборудование.

13.0.4.1

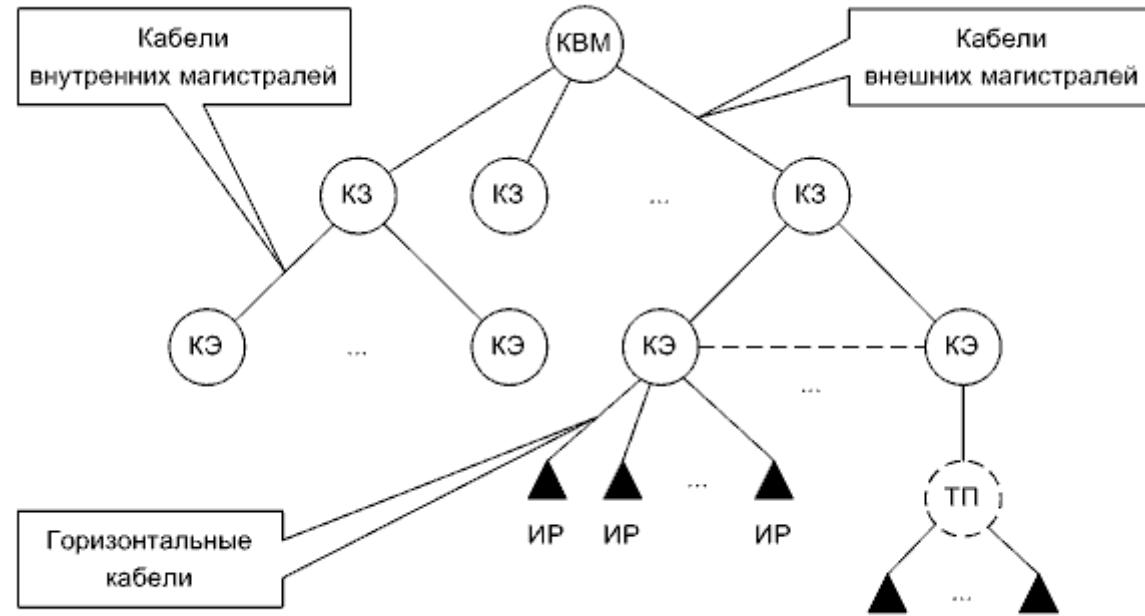
В общем случае, согласно стандартам ISO/IEC 11801, ANSI/TIA/EIA-568 и EN 50173, СКС включает в себя три подсистемы:

1. *Подсистема внешних магистралей* (main, campus) -- основа для организации связи между компактно расположеннымными на одной территории зданиями или сооружениями.

2. *Подсистема внутренних магистралей* или, по-другому, *вертикальная* (intermediate, building) -- связывает между собой этажи одного здания или пространственно разнесенные помещения в одном здании.

3. *Горизонтальная подсистема* (horizontal) -- связывает между собой оборудование в пределах этажа или помещения.

13.0.4.2а



Где:

КВМ -- кроссовая внешних магистралей,

КЗ -- кроссовая здания,

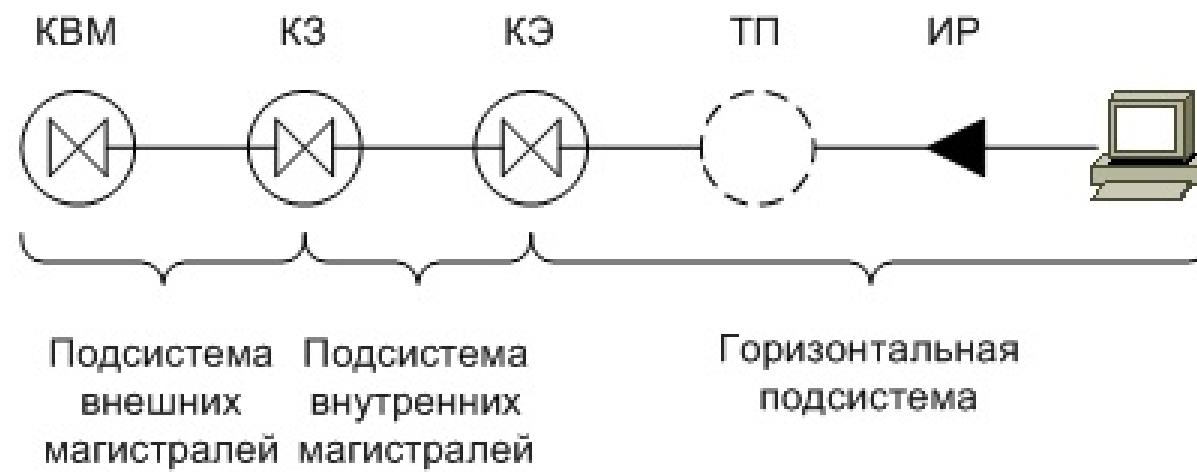
КЭ -- кроссовая этажа,

ТП -- точка перехода,

ИР -- информационная розетка рабочего места.

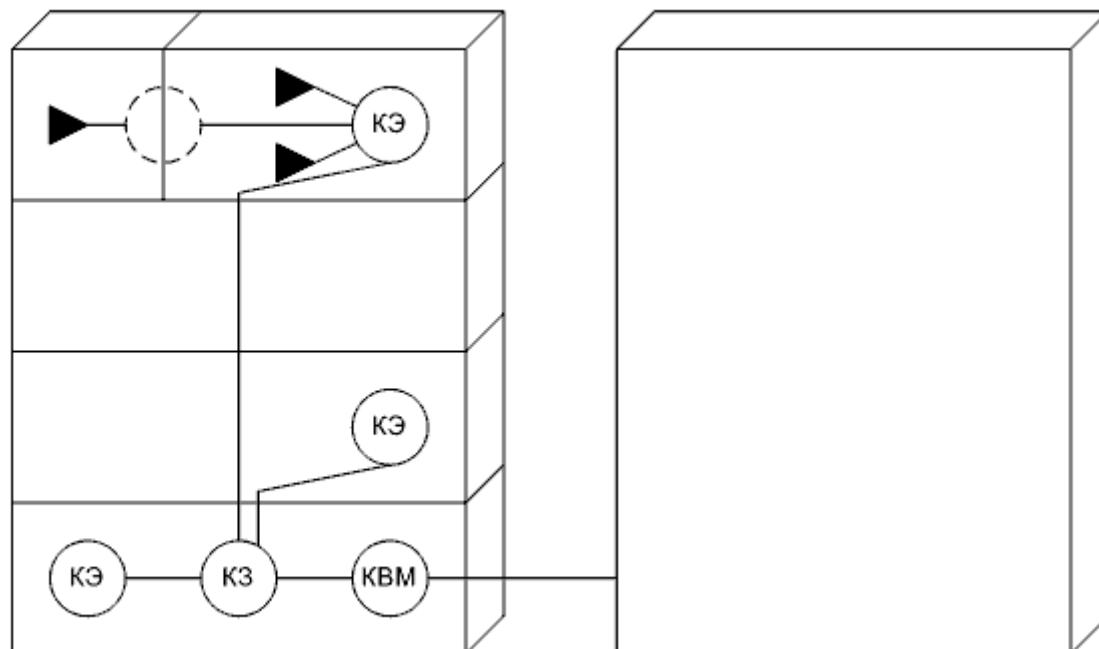
(Вместо кроссовых могут быть аппаратные. Пунктиром обозначены опциональные компоненты. Аббревиатуры -- нестандартные.)

13.0.4.2b



Модели СКС

13.0.4.2с



Модели СКС

13.0.4.3

Таким образом, суммарно СКС содержит: кабели и сетевое оборудование всех трех подсистем, плюс точки перехода (consolidation points), плюс информационные розетки.

Иерархическая сетевая модель Cisco хорошо «ложится» на модели СКС.

13.0.5.1

С точки зрения администрирования СКС выделяют два подхода:

1. *Многоточечный (распределенный).*
2. *Одноточечный (централизованный).*

13.0.5.2

Согласно стандарту ТIA-606 выделяют четыре класса администрации:

Class 1 -- в пределах аппаратной.

Class 2 -- в пределах здания.

Class 3 -- в пределах кампуса.

Class 4 -- за пределами кампуса.

13.0.6.1

При проектировании СКС внимание должно быть уделено подключению к силовым сетям, а также организации защиты посредством заземления, зануления или других способов.

Заземление необходимо для:

1. Предотвращения поражения электрическим током людей.
2. Защиты кабельных трактов и сетевого оборудования как от выхода из строя, так и от помех.
3. Обеспечения возможности прохождения сигналов применительно к некоторым видам сетевого оборудования.

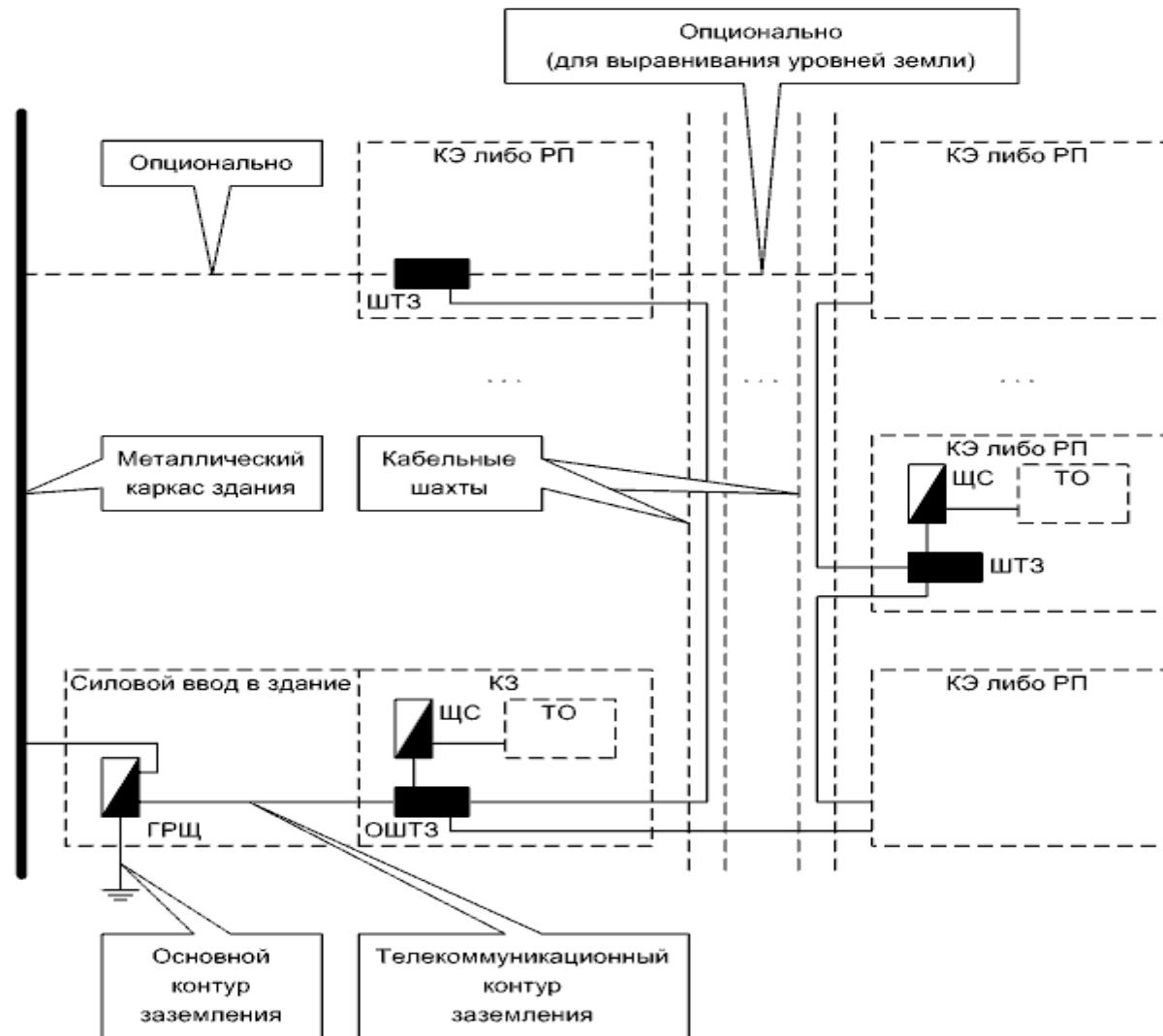
13.0.6.2

Как выполняют заземление?

13.0.6.3

Согласно стандарту TIA-607, в дополнение к основному контуру заземления (grounding electrode) здания либо сооружения, создают так называемый телекоммуникационный контур заземления или, по-другому, контур рабочего заземления (telecommunications grounding/bonding).

13.0.6.4а



Модель телекоммуникационного контура заземления

13.0.6.4b

Где:

ГРЩ -- главный распределительный щит здания,

ЩС -- щит силовой (может быть щит этажный и так далее),

ШТЗ -- шина телекоммуникационного заземления,

ОШТЗ -- основная ШТЗ,

РП -- рабочее помещение,

ТО -- телекоммуникационное оборудование.

(Нарисовано с учетом отечественных особенностей. Аббревиатуры кроме ГРЩ и ЩС -- нестандартные.

Модель телекоммуникационного контура заземления

13.0.6.5

Рекомендации стандартов по заземлению экранов кабелей (касается и витых пар):

1. В аппаратных и кроссовых экраны должны заземляться по возможности на телекоммуникационный контур.
2. Экраны вертикальной подсистемы должны заземляться с обоих концов -- в аппаратных или кроссовых.
3. Экраны горизонтальной подсистемы достаточно заземлять с одного конца -- по возможности в аппаратных или кроссовых.

13.0.7.1

Для защиты от электрических разрядов в атмосфере (особенно вертикальной подсистемы) применяют специальные устройства -- грозоразрядники (lightning gaps).



[DealExtreme]

13.0.8.1

Поскольку современные СКС охватывают здания или сооружения практически полностью, серьезное внимание должно быть уделено и пожарной безопасности.

13.0.8.2

Согласно американским стандартам NEC (National Electrical Code) предусмотрены четыре уровня сертификации пожарной безопасности кабельных систем (первый уровень -- высший):

1. Plenum -- сюда относят кабели, которые можно без каких-либо ограничений прокладывать в так называемых plenum-полостях (существует приток воздуха, достаточный для постоянного горения).
2. Riser -- сюда относят кабели, которые можно прокладывать в кабельных шахтах (например, вертикальных стояках зданий).
3. General purpose -- сюда относят кабели, которые можно без дополнительной защиты прокладывать везде, кроме plenum-полостей и кабельных шахт.
4. Residential (limited use) -- сюда относят кабели, на прокладку которых наложены специфические ограничения (например, только для жилых помещений).

В стандартах IEC 60332, UL 1685, EN 50266 и некоторых других описаны тесты вертикального и горизонтального распространения огня по кабелям.

13.0.9.1

В состав маркировки кабелей часто вводят буквенные обозначения материалов оболочек.

Примеры:

1. PVC (PolyVinyl Chloride) -- ПВХ (поливинил хлорид).
2. PE (PolyEthylene) -- полиэтилен.
3. PA (PolyAmide) -- полиамид (нейлон).
4. FR (Flame Retardant) -- огнестойкий.
5. LS (Low Smoke) -- низкое выделение дыма при горении.
6. NC (Non Corrosive) -- не подвержен коррозии.
7. UVR (Ultra Violet Resistant) -- не подвержен влиянию ультрафиолетового излучения.
8. HF (Halogen Free) = NH (No Halogen) = ZH (Zero Halogen) -- не содержит галогенов.
9. CST (Corrugated Steel Tape armor равно armour) -- бронирован гофрированной стальной лентой.

13.0.10.1

Относительно недавно производители сетевого оборудования стали разрабатывать технологии, позволяющие питать относительно маломощные Ethernet-устройства (например, коммутаторы или точки доступа) через информационные кабели (на основе витых пар), -- технологии под общим названием PoE (Power over Ethernet).

Постепенно были введены два общепромышленных стандарта: 802.3af и 802.3at.

Но до сих пор многие производители используют собственные проприетарные технологии. Примерами могут служить Cisco Universal Power over Ethernet (UPOE) (до 802.3af была еще технология Inline Power), Microsemi PowerDsine (ряд производителей), Passive PoE (ряд производителей).

13.0.10.2

В структуру PoE-системы входит ряд блоков.

PSE (Power Sourcing Equipment) вводит питающее напряжение в кабель.

PD (Powered Device) питается от этого напряжения.

PSE может располагаться либо на конце (одном из двух) кабеля (*endspan*), то есть быть интегрированным в соответствующее сетевое устройство (как правило, мощный коммутатор, подключенный к силовой сети напрямую), либо «вклиниваться» в кабель (*midspan*), то есть быть внешним PoE-инжектором (PoE injector).

Иногда PoE используется и для запитывания «небольших» PD, PoE не поддерживающих, -- со стороны PD в кабель «вклинивается» PoE-DC-адаптер.

13.0.10.3

	802.3af (PoE) (802.3at тип 1)	802.3at (PoE+) (802.3at тип 2)	Cisco (UPOE)
Максимальный выходной ток PSE	0,35 A	0,6 A	1 A
Выходное напряжение PSE	44 – 57 V	50 – 57 V	44 – 57 V
Максимальный ток, потребляемый PD	0,35 A	0,6 A	1 A
Напряжение питания PD	37 – 57 V	47 – 57 V	37 – 57 V
Максимальная мощность PSE	15,4 W	30 W	60 W
Максимальная мощность PD	12,95 W	25,5 W	51 W
Количество задействованных витых пар	2	2	4

Сравнение технологий PoE

13.0.10.4

Исходя из потребляемой мощности, PDs делят на пять стандартных классов:

Class 0. 0,44 -- 12,95 W -- по умолчанию.

Class 1. 0,44 -- 3,84 W -- очень малой мощности.

Class 2. 3,84 -- 6,49 W -- малой мощности.

Class 3. 6,49 -- 12,95 W -- средней мощности.

Class 4. 12,95 -- 25,5 W -- большой мощности.

13.0.10.5

К какому уровню иерархической сетевой модели Cisco относят PoE?

