МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени Н.Э. БАУМАНА

Учебное пособие

Методическое пособие для подготовки к сдаче экзамена по курсу

«Основы телекоммуникационных технологий»

МГТУ имени Н.Э. Баумана

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени Н.Э. БАУМАНА

Методическое пособие для подготовки к сдаче экзамена по курсу

«Основы телекоммуникационных технологий»

Москва
МГТУ имени Н.Э. Баумана

2012

УДК 681.3.06(075.8)
ББК 32.973-018
И201

Методическое пособие для подготовки к сдаче экзамена по курсу «Основы телекоммуникационных технологий» / Коллектив авторов – М.: МГТУ им. Н.Э. Баумана, 2012. – 73 с.: ил.

В методическом пособии были рассмотрены основные этапы курса «Основы телекоммуникационных технологий».

Ил. 39. Табл. 5. Библиогр. 7 назв.

УДК 681.3.06(075.8)

АННОТАЦИЯ

В методическом пособии рассмотрены основные темы курса «Основы телекоммуникационных технологий» такие как: сети, основы их посмотрения и передачи данных.

ANNOTATION

The course of lectures addressed the main themes of the course "Basis of telecommunication technology" such as webs anb basis of their building and transferring information via networks.

1. ВОПРОСЫ

по курсу "Основы телекоммуникационных технологий"

Раздел 1

1. Сетевые топологии: понятие, сравнительные характеристики.
2. Понятие о кластеризации: основные определения и термины. Классификация. Сферы применения.
3. Модель OSI.
4. Архитектура систем хранения данных SAN
5. Основные устройства физического уровня модели OSI и их характеристики
6. Ethernet. Особенности физической реализации.
7. Протокол FCIP iFCIP.
8. Технологии канального уровня и модель сетевой организации. Понятия инкапсуляции, конвергенции и туннелирования.
9. Клиент серверное взаимодействие. Виды соединий. Понятие широковещательной сети.
10. Проектирование сетей: домены коллизий.
11. Проектирование сетей: Понятие СКС, основные конструктивы, методы монтажа, ограничения.
12. Проектирование сетей: трассировка кабельных трасс.
13. Проектирование сетей: Концепция сетевой безопасности: аутентификация, целостность сообщений, конфиденциальность с помощью симметричного шифрования, ассиметричный общедоступный ключ шифрования, комбинированное шифрование.
14. Протокол PPP: характеристики, сжатие в PPP, аутентификация, автоматичсекое отслеживание качества связи.
15. Конфигурация сетей с помощью BOOTP и DHCP.
16. Протоколы Ethernet. Общие понятие, определения и термины, особенности.
17. Сетевые службы и сервисы. Понятие и основные характеристики.
18. Протокол NETBIOS
19. Протоколы транспортного уровня (TCP, UDP).
20. Понятие “socket”. Службы, вызовы, принципы работы.
21. Представление FCP по уровням модели OSI (физический уровень, кодирование передаваемой информации, контроль канала передачи, сервис передачи данных, FC-4).
22. Понятие о GRID технологиях.
23. Сетевые антивирусные средства. Классификация, приницпы работы.

Раздел 2

1. Протокол ICMP. Модель, основные команды, безопасность, производительность.
2. Сетевые архитектуры: понятие, сравнительные характеристики.
3. Протокол POP. Модель, основные команды, безопасность, производительность.
4. Конвергенция сетей.
5. Виды и характеристики физических каналов передачи данных
6. Сети Frame Relay.
7. Маршрутизация: маршрутизация первого уровня.
8. Сети ATM.
9. Маршрутизация: маршрутизация первого уровня.
10. Протокол SMTP. Модель, основные команды, безопасность, производительность.
11. Протокол маршрутизации EGP
12. Протокол маршрутизации RIP.
13. Сети X.25.
14. Понятие MAC адреса, его структура.
15. Протокол маршрутизации BGP

16. IP адресация: IPv4, IPv6. Варианты назначения IP адресов.
17. Протокол маршрутизации OSPF
18. Маршрутизация: маршрутизация второго уровня.
19. Понятие маски подсети, ее назначение. Безклассовая модель представления сетевых адресов.
20. Маршрутизация: маршрутизация третьего уровня.
21. Разрешение сетевых имен с помощью DNS. Протокол ARP.
22. Понятие фреймов Ethernet (IEEE 802.3 Packet Framing), изменения в Ethernet II.
23. Протоколы маршрутизации: RIP, OSPF, BGP, EGP. Сравнительные характеристики.
24. Протокол SLIP.
25. Понятие пакета, его структура. Технологии передачи пакетов в Ethernet.
26. Маршрутизация: основные понятия, уровни маршрутизации.
27. Протокол FTP. Модель, основные команды, безопасность, производительность.
28. NFS, RPC и XDR.

Раздел 3

1. Сетевые системы хранения данных: Протокол Serial ATA.
2. Сетевые системы хранения данных: Протокол Parallel ATA.
3. Сетевые системы хранения данных: Протокол iSCSI.
4. Сетевые системы хранения данных: Протокол Parallel SCSI.
5. Архитектура систем хранения данных NAS
6. Сетевые системы хранения данных: Дисковые массивы: JBOD, RAID.
7. Архитектура систем хранения данных DAS
8. Сетевые системы хранения данных: Оптические и магнито-оптические устройства хранения данных.
9. Архитектуры систем хранения данных: Сравнительные характеристики DAS, NAS, SAN, рекомендации по применению.
10. Сетевые системы хранения данных: Дисковые массивы с RAID: уровни RAID, принципы организации по уровням.
11. Сети хранения данных – основные понятия, определения и термины. Дисковые устройства хранения данных.
12. Сети хранения данных – основные понятия, определения и термины. Ленточные устройства хранения данных.
13. Протокол FCP (Fibre Channel Protocol)

Пример задачи:

1. Разработать bat (batch) файл. Содержащий команды управления, которые позволяют вывести на экран имя и адрес локального сервера разрешения сетевых имен (привести листинг с типовыми ответами сервера).
2. Разработать bat (batch) файл, содержащий команды управления, которые позволяют получить параметры конфигурации хоста, включая IP-адрес, маску подсети и шлюз по умолчанию, отобразить полную информацию о настройке параметров, освободить IP-адрес для указанного адаптера, обновить IP-адрес для указанного адаптера, очистить кэш разрешений (DNS), обновить все DHCP-аренды и перерегистрировать DNS-имена, отобразить содержимое кэша разрешений (DNS), отобразить все допустимые для этого адаптера коды (IDs) DHCP-классов, изменить код (ID) DHCP-класса (привести листинг с типовыми ответами сервера).

2. ЗАДАЧИ

3. **Разработать bat (batch) файл. Содержащий команды управления, которые позволяют вывести на экран имя и адрес локального сервера разрешения сетевых имен (привести листинг с типовыми ответами сервера).**

test.bat:
nslookup .

Типовой ответ:
Server: h129.net37.bmstu.ru
Address: 195.19.37.129

Приведите последовательность команд для копирования файла tese.txt из каталога book сервера iu4.bmstu.ru на локальный хост с использованием FTP (login: test, password:test). (привести листинг с типовыми ответами сервера)

Зашли в ftp

open iu4.bmstu.ru
> USER test
> PASS test
cd book
lcd C:/get_file_in_here
get tese.txt
…
disconnect
quit

1. **Разработать bat (batch) файл, содержащий команды управления, которые позволяют**

получить параметры конфигурации хоста, включая IP-адрес, маску подсети и шлюз по умолчанию, отобразить полную информацию о настройке параметров, освободить IP-адрес для указанного адаптера, обновить IP-адрес для указанного адаптера, очистить кэш разрешений (DNS), обновить все DHCP-аренды и перерегистрировать DNS-имена, отобразить содержимое кэша разрешений (DNS), отобразить все допустимые для этого адаптера коды (IDs) DHCP-классов, изменить код (ID) DHCP-класса (привести листинг с типовыми ответами сервера).

Делается через Ipconfig

/all	Отобразить полную информацию о настройке параметров.
/release	Освободить IP-адрес для указанного адаптера.
/renew	Обновить IP-адрес для указанного адаптера.
/flushdns	Очистить кэш разрешений DNS.
/registerdns	Обновить все DHCP-аренды и перерегистрировать DNS-имена
/displaydns	Отобразить содержимое кэша разрешений DNS.

/showclassid	Отобразить все допустимые для этого адаптера коды (IDs) DHCP-классов.
/setclassid	Изменить код (ID) DHCP-класса.

1. **Разработать bat (batch) файл, содержащий команды управления, которые позволяют**

проверить возможность отправки пакетов на указанный узел до команды прерывания и выполнить: определение адресов по именам узлов, установить число отправляемых запросов, размер буфера отправки, установить флаг, запрещающего фрагментацию пакета, задать срока жизни пакета, задать тип, обеспечить запись маршрута для указанного числа переходов, задать штамп времени для указанного числа переходов, обеспечивать свободный или жесткий выбор маршрута по списку узлов, задать таймаут каждого ответа в миллисекундах (привести листинг с типовыми ответами сервера).

Выполняется командой ping:

Параметры:

-t Отправка пакетов на указанный узел до команды прерывания.
Для вывода статистики и продолжения нажмите
<Ctrl>+<Break>, для прекращения - <Ctrl>+<C>.
-a Определение адресов по именам узлов.
-n число Число отправляемых запросов.
-l размер Размер буфера отправки.
-f Установка флага, запрещающего фрагментацию пакета.
-i TTL Задание срока жизни пакета (поле "Time To Live").
-v TOS Задание типа службы (поле "Type Of Service").
-r число Запись маршрута для указанного числа переходов.
-s число Штамп времени для указанного числа переходов.
-j списокУзлов Свободный выбор маршрута по списку узлов.
-k списокУзлов Жесткий выбор маршрута по списку узлов.
-w таймаут Таймаут каждого ответа в миллисекундах.

1. **Разработать bat (batch) файл, содержащий команды управления, которые позволяют**

провести трассировку маршрута до запрашиваемого узла (платформа Win32): без разрешения в имена узлов, с указанием максимального число прыжков при поиске узла, формированию свободный выбор маршрута по списку узлов, заданию интервала ожидания каждого ответа в миллисекундах (привести листинг с типовыми ответами сервера).

Программа tracert (для Линукса это traceroute)

Параметры:

-d Без разрешения в имена узлов.
-h максЧисло Максимальное число прыжков при поиске узла.
-j списокУзлов Свободный выбор маршрута по списку узлов.
-w интервал Интервал ожидания каждого ответа в миллисекундах.

1. **Разработать bat (batch) файл, содержащий команды управления, которые позволяют**

сформировать и отображать таблицы сетевых маршрутов и выполнять: очистку таблиц маршрутов от записей для всех шлюзов, печать маршрута, добавление маршрута, удаление маршрута, изменение существующего маршрута, задание адресуемого узла, задание маски подсети, связываемое с записью для данного маршрута, указание шлюза, определение параметра метрика/цена для адресуемого узла (привести листинг таблицы маршрутизации до и после изменений).

ROUTE [-f] [-p] [команда [узел]
[MASK маска] [шлюз] [METRIC метрика] [IF-интерфейс]

-f Очистка таблиц маршрутов от записей для всех шлюзов. При указании одной из команд, таблицы очищаются до выполнения команды.

-p При использовании с командой ADD задает сохранение маршрута при перезагрузке системы. По умолчанию маршруты не сохраняются при перезагрузке. Игнорируется для остальных команд изменяющих соответствующие постоянные маршруты. Этот параметр не поддерживается в Windows 95.

команда Одна из четырех команд
PRINT Печать маршрута
ADD Добавление маршрута
DELETE Удаление маршрута
CHANGE Изменение существующего маршрута

узел Адресуемый узел.

MASK Если вводится ключевое слово MASK, то следующий параметр интерпретируется как параметр "маска".

маска Значение маски подсети, связываемое с записью для данного маршрута. Если этот параметр не задан, по умолчанию подразумевается 255.255.255.255.

шлюз Шлюз.

METRIC Определение параметра метрика/цена для адресуемого узла.

Поиск всех символических имен узлов проводится в файле сетевой базы данных NETWORKS. Поиск символических имен шлюзов проводится в файле базы данных имен узлов HOSTS.

Для команд PRINT и DELETE можно указать узел и шлюз с помощью подстановочных знаков или опустить параметр "шлюз".

Если адресуемый узел содержит подстановочные знаки * или ?, он используется в качестве шаблона, и печатаются только соответствующие ему маршруты.

1. **Разработать bat (batch) файл, содержащий команды управления, которые позволяют**

сформировать таблицу статической маршрутизации (привести листинг таблицы маршрутизации до и после изменений).

test.bat:

```
route ADD 192.168.58.126 192.168.58.1 METRIC 10 IF 2
route ADD 192.168.58.76 192.168.58.1 METRIC 10 IF 2
route ADD 192.168.58.226 MASK 255.255.255.255 192.168.58.226 METRIC 10 IF 2
```

Таким образом добавили 2 адреса (126 и 76) с маской по умолчанию 255.255.255.255, шлюзом 192.168.58.1, метрикой 10 на интерфейсе 2 (как правило, это наш сетевой Ethernet-адаптер). Третий адрес (226) замкнули сам на себя.

1. **Разработать топологию сети, которая не включает в себя «домен коллизий».**

Во-первых, нельзя использовать концентраторы.
Во-вторых, даже используя свитчи (коммутаторы), все равно коллизии возможны (это связано с тем, что когда несколько активных источников посылают много данных одному получателю, плохой коммутатор (с малым буфером) не успевает сохранить все приходящие данные и они теряются).
Выход – использовать маршрутизаторы.

1. **Разработать bat (batch) файл, содержащий команды управления, которые позволяют**

сформировать таблицу маршрутизации для всех хостов заданной подсети (привести листинг таблицы маршрутизации до и после изменений).

```
route ADD 192.168.58.0 MASK 255.255.255.0 192.168.58.1 METRIC 10 IF 1
```

Этой записью мы указали, что наш маршрутизатор будет перенаправлять все пакеты с IP получателя 192.168.58.1 – 192.168.58.254 на следующий маршрутизатор 192.168.58.1, с метрикой 10, через сетевой интерфейс 1

1. **Разработать bat (batch) файл, содержащий команды управления, которые позволяют**

Просмотр локальной ARP таблицы, ручное удаление и добавление элементов таблицы, загрузку информации в таблицу из конфигурационного файла

Это все делается через arp. Вот описание:

```
Отображение и изменение таблиц преобразования IP-адресов в физические, использу
мые протоколом разрешения адресов (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr]

  -a            Отображает текущие ARP-записи, опрашивая текущие данные
                протокола. Если задан inet_addr, то будут отображены IP и
                и физический адреса только для заданного компьютера.
                Если более одного сетевого интерфейса используют ARP,
                то будут отображаться записи для каждой таблицы.
  -g            То же, что и ключ -a.
  inet_addr     Определяет IP-адрес.
  -N if_addr    Отображает ARP-записи для заданного в if_addr сетевого
                интерфейса.
  -d            Удаляет узел, задаваемый inet_addr. inet_addr может содержать
                символ шаблона * для удаления всех узлов.
  -s            Добавляет узел и связывает internet адрес inet_addr
                с физическим адресом eth_addr. Физический адрес задается
                6 байтами (в шестнадцатеричном виде), разделенных дефисом.
                Эта связь является постоянной.
  eth_addr      Определяет физический адрес.
  if_addr       Если параметр задан, - он определяет интернет адрес интерфейса,
                чья таблица преобразования адресов должна измениться.
                Если не задан, - будет использован первый доступный интерфейс.
Пример:
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  ... Добавляет статическую запись.
  > arp -a                                    ... Выводит ARP-таблицу.
```

Для того, чтобы загрузить инфу из кофиг. файла (под Windows), потребуется действительно сделать bat-файл:

FOR /F "tokens=1,2" %%i in (%1) do @ARP -s %%i %%j .

Здесь в %1 задан параметр – файл, который надо считать

Под Линуксом это делается автоматом:

arp –f “~/my_config_file”

Разработать bat (batch) файл, содержащий команды управления, которые позволяют с помощью команды netstat получить информацию о маршруте по умолчанию, о интерфейсах на который направляются датаграммы, о статистике по местному трафику, количесвте активных применений маршрута, о кольцевом трафике, по оценке пригодности маршрута для использования.

netstat –e - статистика по местному траффику (статистика интерфейса)
netstat -r - все остальное (эта команда аналогична route print)

Вообще, эта задача какая-то мутная, еще тупее остальных. Что тут подразумевается под маршрутом, не ясно…

1. Провести верификацию типового ответа сервера на команду netstat -s

Типовый ответ (верифицируйте ☺):

Статистика IPv4

Получено пакетов = 551076
Получено ошибок в заголовках = 0
Получено ошибок в адресах = 14658
Направлено датаграмм = 0
Получено неизвестных протоколов = 0
Отброшено полученных пакетов = 20
Доставлено полученных пакетов = 550726
Запросов на вывод = 455804
Отброшено маршрутов = 0
Отброшено выходных пакетов = 0
Выходных пакетов без маршрута = 0
Требуется сборка = 0
Успешная сборка = 0
Сбоев при сборке = 0
Успешно фрагментировано датаграмм = 0
Сбоев при фрагментации датаграмм = 4
Создано фрагментов = 0

Статистика ICMPv4

	Получено	Отправлено
Сообщений	337	344
Ошибок	0	0
'Назначение недостижимо'	5	54
Превышений времени	30	0
Ошибок в параметрах	0	0
Просьб "снизить скорость"	0	0
Переадресовано	71	0
Эхо-сообщений	119	171
Ответных пакетов	112	119
Штампов времени	0	0
Ответы на штампы времени	0	0
Масок адресов	0	0
Ответов на маски адресов	0	0

Статистика TCP для IPv4

Активных открыто = 14154
Пассивных открыто = 31590
Сбоев при подключении = 5420
Сброшено подключений = 13880
Текущих подключений = 3
Получено сегментов = 456039
Отправлено сегментов = 440426
Повторно отправлено сегментов = 48

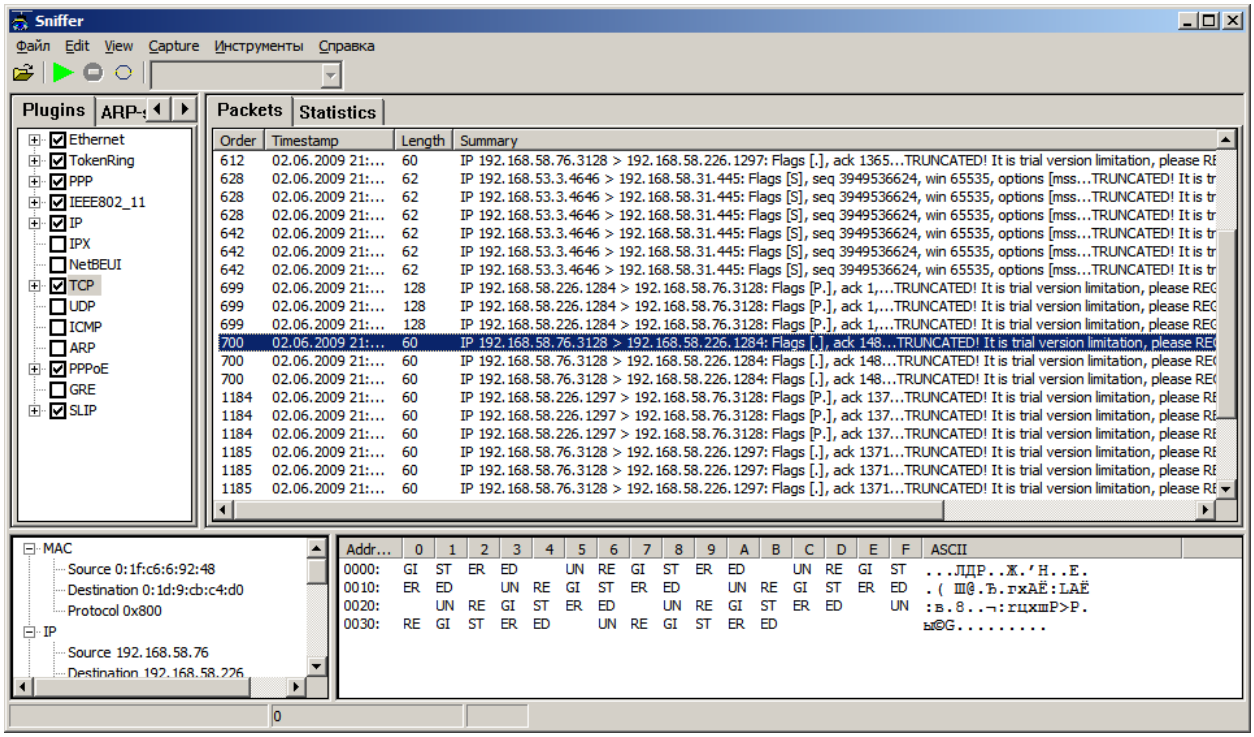
Статистика UDP для IPv4

Получено датаграмм = 93640
Отсутствие портов = 1632
Ошибки при получении = 0
Отправлено датаграмм = 14988

Привести пример работы сетевого анализатора Sniffer , который представляет информацию о сегментах TCP.

Анализатор трафика, или снифф‌ер (от англ. to sniff — нюхать) — сетевой анализатор трафика, программа или программно-аппаратное устройство, предназначенное для перехвата и последующего анализа, либо только анализа сетевого трафика, предназначенного для других узлов.
Во время работы снифф‌ера сетевой интерфейс переключается в т.н. «режим прослушивания» (Promiscuous mode), что и позволяет ему получать пакеты, адресованные другим интерфейсам в сети.

Скриншотик снифф‌ера:



1. **Провести верификацию ответа программы netstat –an, которая позволяет осуществить**

Проверку текущего состояния соединения.

Имя	Локальный адрес	Внешний адрес	Состояние
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1110	0.0.0.0:0	LISTENING
TCP	0.0.0.0:19780	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1029	0.0.0.0:0	LISTENING
TCP	127.0.0.1:4474	127.0.0.1:1110	TIME_WAIT
TCP	127.0.0.1:4477	127.0.0.1:1110	TIME_WAIT
TCP	192.168.58.226:139	0.0.0.0:0	LISTENING
TCP	192.168.58.226:1284	192.168.58.76:3128	ESTABLISHED
TCP	192.168.58.226:1297	192.168.58.76:3128	ESTABLISHED
TCP	192.168.58.226:3784	192.168.58.76:445	ESTABLISHED
TCP	192.168.58.226:4473	192.168.58.76:3128	TIME_WAIT
TCP	192.168.58.226:4476	192.168.58.76:3128	TIME_WAIT
TCP	192.168.58.226:4480	192.168.58.208:139	TIME_WAIT
TCP	192.168.58.226:4482	192.168.58.208:139	ESTABLISHED
UDP	0.0.0.0:445	*:*	
UDP	0.0.0.0:500	*:*	
UDP	0.0.0.0:1184	*:*	
UDP	0.0.0.0:4500	*:*	
UDP	127.0.0.1:123	*:*	
UDP	127.0.0.1:1138	*:*	
UDP	127.0.0.1:1900	*:*	
UDP	127.0.0.1:4030	*:*	
UDP	192.168.58.226:123	*:*	
UDP	192.168.58.226:137	*:*	
UDP	192.168.58.226:138	*:*	
UDP	192.168.58.226:1900	*:*	

1. **Провести верификацию структуры конфигурационного файла данных DNS для прямого разрешения имен**

Под Линуксом это файл **/var/named/private.hosts**
Вот что там содержится:

```
;name    ttl    class    type    data
localhost       IN       A       127.0.0.1
solaris         IN       A       10.5.3.1
win95           IN       A       10.5.3.21
linux           IN       A       10.5.3.22

;
;    Aliases
;
mail            IN       CNAME   solaris
www             IN       CNAME   solaris
```

1. **Провести верификацию структуры конфигурационного файла данных DNS для обратного разрешения имен**

Под Линуксом это файл **/var/named/private.rev**
Вот что там содержится:

```
;name    ttl    class    type    data
1               IN    PTR    solaris.flibble.orac.net.au.
21              IN    PTR    win95.flibble.orac.net.au.
22              IN    PTR    linux.flibble.orac.net.au.
```

Управление текстовым клиентом telnet, основные команды и важные управляющие последовательности

open имя_ЭВМ [порт]	open открывает связь с ЭВМ, имя которой указано в обращении. Если номер порта явно не указан, telnet пытается использовать для связи с сервером номер порта по умолчанию. Вместо имени ЭВМ-сервера может использоваться ее IP-адрес.
display [аргумент ...]	Отображает все, или часть, набора параметров telnet (см. описание команды send).
close	Закрывает сессию telnet и возвращает систему в командный режим.
quit	Закрывает любую сессию telnet.
mode type	Управляет режимом ввода ("построчный" или "посимвольный"). Удаленной машине посылается запрос на переход в соответствующий режим. Если она готова (способна) работать в запрошенном режиме, будет произведено соответствующее переключение.
status	Отображает текущий статус telnet. В перечень информации входит имя удаленной ЭВМ и действующий режим обмена.
? [команда]	Выдает справочную информацию о команде, название которой приведено в качестве аргумента
send arguments	Посылает удаленной ЭВМ один или несколько символьных аргументов. В качестве аргументов могут использоваться: escape, synch, brk, ip, ao, ayt, ecel, ga и др. Смотри таблицу 4.5.3.3.
escape	Посылает escape символ (например, `^]').
SYNCH	Посылает synch-последовательность. Эта последовательность позволяет аннулировать все, что было до этого напечатано, но еще не считано. Эта последовательность посылается как срочная (важная) TCP-информация (может не сработать, если удаленной системой является 4.2 BSD). Если она не сработала, на терминал будет послан символ "r".
brk	Посылает Break-последовательность при нажатии клавиши Break (Pause). (Исчерпывающую информацию об аргументах можно найти в описании используемого программного обеспечения или с помощью команд Help или Man)
set argument value	Присваивает любому числу переменных telnet новые значения. Специальное значение "off" выключает функцию, соответствующую данной переменной

Последовательность символов	Назначение
?	Отображает справочную информацию о команде**send**
escape	Посылает символ escape (без прерывания посылки символов для Telnet)
ip	Посылает протокольную последовательность telnet. Удаленная машина должна прервать процесс, запущенный для вас.
ec	Посылает протокольную EC-последовательность telnet. Удаленная ЭВМ должна стереть последний напечатанный вами символ
el	Посылает протокольную EL-последовательность TELNET. Удаленная ЭВМ должна стереть последнюю напечатанную вами строку.
ao	Посылает протокольную AO-последовательность TELNET. Удаленная ЭВМ должна направить весь вывод на ваш терминал.
brk	Посылает протокольную BRK-последовательность TELNET. Удаленная ЭВМ должна обеспечить отклик.
ayt	Посылает протокольную AYT-последовательность TELNET (Are You There). Удаленная ЭВМ должна обеспечить отклик.

1. Использование команды netstat.

Отображение статистики протокола и текущих сетевых подключений TCP/IP.

NETSTAT [-a] [-b] [-e] [-n] [-o] [-p протокол] [-r] [-s] [-v] [интервал]

-a Отображение всех подключений и ожидающих портов.

-b Отображение исполняемого файла, участвующего в создании каждого подключения, или ожидающего порта. Иногда известные исполняемые файлы содержат множественные независимые компоненты. Тогда отображается последовательность компонентов, участвующих в создании подключения, либо ожидающий порт. В этом случае имя исполняемого файла находится снизу в скобках [], сверху - компонент, который им вызывается, и так до тех пор, пока не достигается TCP/IP. Заметьте, что такой подход может занять много времени и требует достаточных разрешений.

-e Отображение статистики Ethernet. Он может применяться вместе с параметром -s.

-n Отображение адресов и номеров портов в числовом формате.

-o Отображение кода (ID) процесса каждого подключения.

-p протокол Отображение подключений для протокола, задаваемых этим параметром. Допустимые значения: TCP, UDP, TCPv6 или UDPv6. Используется вместе с параметром -s для отображения статистики по протоколам. Допустимые значения: IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP или UDPv6

-r Отображение содержимого таблицы маршрутов.

-s Отображение статистических данных по протоколам. По умолчанию данные отображаются для IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP и UDPv6. Параметр -p позволяет указать подмножество выводящихся данных.

-v При использовании с параметром -b, отображает последовательность

компонентов, участвующих в создании подключения, или ожидающий порт для всех исполняемых файлов.

интервал Повторный вывод статистических данных через указанный промежуток времени в секундах. Для прекращения вывода данных нажмите клавиши CTRL+C. Если параметр не задан, сведения о текущей конфигурации выводятся один раз.

Раздел 1

1. Сетевые топологии: понятие, сравнительные характеристики.

Данной статьей мы открываем раздел «В помощь системному администратору». В статье рассмотрены наиболее общие вопросы сетевых технологий. Для специалистов по сетям статья не представляет какого-либо интереса, непосвященным же позволит познакомиться с основными терминами и понятиями.

Сетью (network) называется группа соединенных компьютеров (и других устройств).

Объединение компьютеров в сети позволяет решать следующие задачи:

- совместное использование информации (например, файлов);
- совместное использование аппаратных средств (например, принтера, модема и др.);
- совместное использование программных ресурсов (например, программы типа клиент-сервер);
- обеспечение единой политики безопасности для узлов сети (например, настройка безопасности рабочих станций на сервере при подключении локальной сети к Интернет);
- разграничение полномочий узлов сети (например, для распределения полномочий между различными подразделениями предприятия);
- обеспечение защиты информации совместного использования (например, резервное копирование на стороне сервера);
- обеспечение обмена данными между узлами сети (например, при использовании электронной почты).

Классификация сетей

Существует множество классификаций сетей, проводимых по различным критериям. Рассмотрим некоторые из них.

По распределению полномочий компьютеров сети можно разделить на *одноранговые, серверные и гибридные*.

В **одноранговых сетях** все компьютеры имеют одинаковые «права и обязанности». Каждый компьютер предоставляет свои ресурсы другим членам сети и одновременно может пользоваться их ресурсами.

В **серверных сетях** один или несколько компьютеров (серверы) предоставляют свои ресурсы всем другим компьютерам сети (клиентам). При этом сервер не использует ресурсы клиентов.

В **гибридных сетях** совмещены признаки одноранговых и серверных сетей. Например, один узел, будучи сервером для части компьютеров, может являться клиентом другого сервера.

По числу подключенных к сети узлов, а также их географическому расположению сети делятся на *локальные, региональные и глобальные*. Большинство сетей являются гибридными.

Локальные сети (LAN - Local Area Networks, ЛВС - Локальные Вычислительные Сети) представляют собой несколько компьютеров, имеющих общую среду передачи данных, и физически расположенных близко друг от друга (например, в одном здании или комнате). Физическая близость компьютеров :) в локальных сетях позволяет использовать в LAN технологии, поддерживающие передачу данных на чрезвычайно высоких скоростях.

Региональные сети (MAN - Metropolitan Area Network) - представляют собой несколько сот, тысяч или более компьютеров, расположенных на относительно удаленном расстоянии друг от друга (например, в пределах одного города или области) и имеющих при этом общую среду передачи данных.
Региональные сети работают на скоростях от средних до высоких.

Глобальные сети (WAN - Wide Area Network)- это совокупность региональных сетей, связанных коммуникационными каналами.
В качестве коммуникационных каналов чаще всего используются телефонные линии, а также более дорогие варианты: оптоволононные кабели, спутниковые каналы и др.
Глобальные сети работают на самых низких скоростях передачи данных.
Примером глобальной сети служит сеть Интернет.

Сетевые топологии

Набор правил для физического соединения узлов сети и организации взаемодействия сетевых устройств называется **сетевой топологией**.
Конфигурация физических связей определяется электрическими соединениями узлов сети между собой и может отличаться от конфигурации логических связей. **Логические связи** представляют сбой маршруты передачи данных между узлами сети и образуются путем соответствующей настройки коммуникационного оборудования.

Топологии сетей можно разделить на две основные группы: *полносвязные и неполносвязные* (рис. 1).

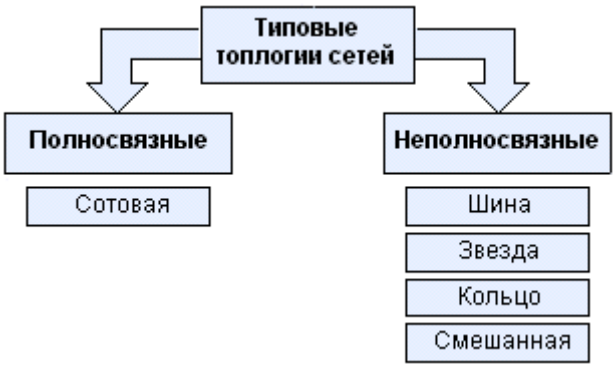


Рис. 1. Топологии сетей.

В **сети с полносвязной топологией** каждый компьютер сети напрямую связан с каждым компьютером этой сети (рис. 2).
Примером такой сети является сеть ячеистой (сотовой) топологии.

В некоторых литературных источниках сеть с неполной сотовой структурой (с отсутствием одной или нескольких связей) называют ячеистой. В данном случае в такие подробности мы вдаваться не будем.

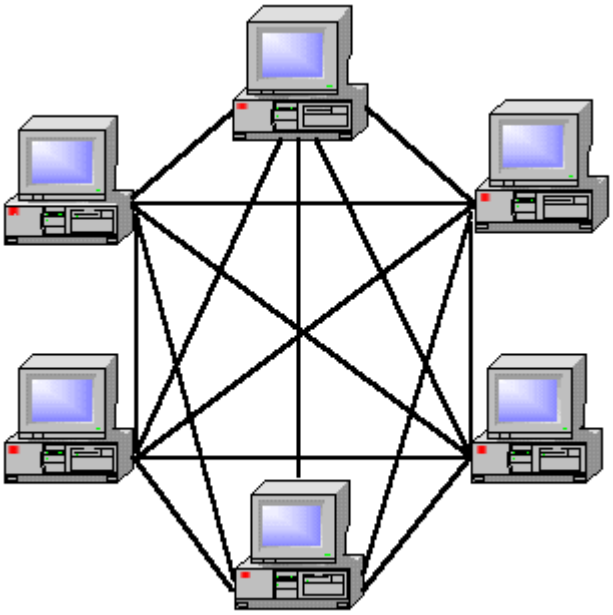


Рис. 2. Сеть сотовой топологии.

Преимущества сотовых сетей:

- Высокая надежность, обусловленная избыточностью физических связей.
- простота диагностики.

Недостатки сотовых сетей:

- Необходимость наличия у каждого компьютера сети большого числа коммуникационных портов для соединения со всеми другими компьютерами.
- Необходимость выделения отдельной электрической линии связи для каждой пары компьютеров.
- Вышеперечисленное обуславливает высокую стоимость сотовой сети.
- Сложность инсталляции и реконфигурации добавления или удаления новых узлов).

Большинство сетевых топологий имеет неполносвязную структуру. К основным видам неполносвязных топологий можно отнести: шину, звезду, кольцо и смешанная топология.

Сети шинной топологии

В сетях с шинной топологией каждый компьютер сети подключен к одному общему кабелю (рис. 3).

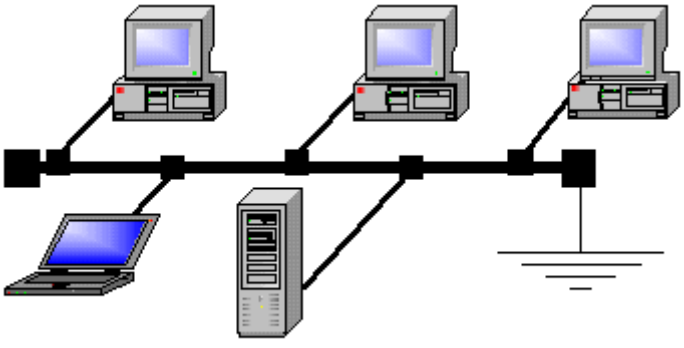


Рис. 3. Сеть шинной топологии.

В шинной топологии отсутствуют активные схемы передачи сигнала от одного компьютера к другому. Когда одна из машин посылает сигнал, он свободно путешествует по всей длине кабеля. Достигнув конца кабеля, сигнал отражается и идет в обратном направлении (зацикливание). Для предотвращения зацикливания сигнала в сетях с шинной топологией обязательно использование оконечной нагрузки (терминатора) на обоих концах кабеля. Сигнал, посланный одной машиной, получают все компьютеры, подключенные к шине. Принимает же его только машина, адрес которой совпал с адресом получателя, закодированном в сообщении.
В каждый момент времени только один из компьютеров может передавать сигнал, остальные должны ждать своей очереди. Соответственно, пропускная способность сетей с шинной топологией невелика и ограничивается не только характеристиками кабеля, но и логической структурой сети.

Достоинства шинной топологии:

- Низкая стоимость.
- Простота расширения (простота подключения новых узлов и объединения двух подсетей с помощью повторителя).

Недостатки шинной топологии:

- Низкая производительность.
- Низкая надежность (частые дефекты кабелей и разъемов).
- Сложность диагностики при разрыве кабеля или отказе разъема.
- Любой дефект кабеля или разъема приводит к неработоспособности всей сети.

Из всего вышесказанного можно заключить, что шинная топология может применяться при небольшом числе узлов в сети и невысокой степени взаимодействия между ними. Вместе с тем, такая сеть отличается низкой стоимостью.

Звездообразная топология

В сетях звездообразной топологии (рис. 3) каждый узел подключается отдельным кабелем к общему устройству, называемому концентратором (хабом). Концентратор передает данные от одного компьютера другому или всем остальным компьютерам сети.

Рис. 4. Сеть звездообразной топологии.

Топология звезда позволяет использовать для подключения компьютеров различные типы кабелей. Наличие концентратора чаще всего делает возможным использование нескольких типов кабелей одновременно.

Достоинства звездообразной топологии:

- Более высокая пропускная способность по сравнению с шинной топологией.
- Выход из строя одного узла или нескольких узлов не влияет на работоспособность остальной сети.
- Легкость включения в сеть новых узлов.
- Возможность использования вместо хаба коммутатора (для фильтрации траффика, а также для мониторинга сети).
- Возможность использования в одной сети нескольких типов кабелей.
- Легкость создания подсетей путем приобретения дополнительного концентратора, подсоединения к нему машин и соединения концентраторов между собой.

Недостатки звездообразной топологии:

- Ограниченная возможность увеличения числа узлов сети (ограничивается количеством портов концентратора).
- Зависимость работоспособности сети от состояния концентратора.
- Высокий расход кабеля (отдельный кабель для подключения каждого компьютера).
- Более высокая стоимость по сравнению с шинной топологей (затраты на хаб и кабель).

Таким образом, сети звездообразной топологии целесообразно прокладывать в зданиях (помещениях), в которых от каждого компьютера можно проложить кабель до концентратора. При планировании такой сети особое внимание следует уделить выбору концентратора.

Кольцевая топология

В сетях с кольцевой топологией (рис. 5) каждый компьютер подключается к общему сетевому кабельному кольцу, по которому передаются данные (в одном направлении).

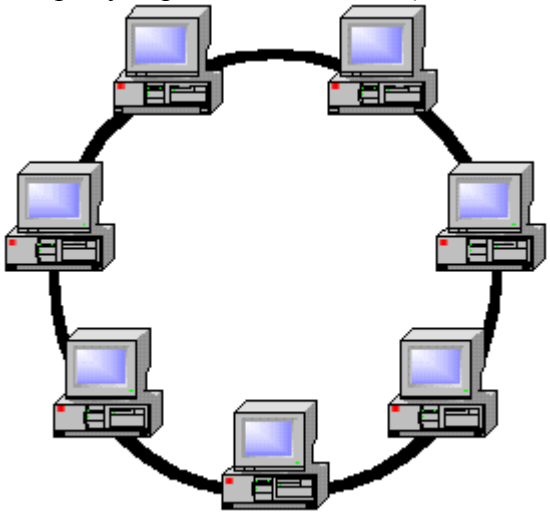


Рис. 5. Сеть кольцевой топологии.

Каждый компьютер, получив данные, сверяет адрес получателя с собственным и в случае из совпадения копирует данные в свой внутренний буфер. Сами данные при этом продолжают движение по кольцу и возвращаются к отправителю. Если, получив данные, компьютер обнаружил, что его адрес не совпадает с адресом получателя, он ретранслирует данные следующему компьютеру в кольце.

В качестве среды передачи данных для построения сети кольцевой топологии чаще всего используют экранированную или неэкранированную «витую пару», а также оптоволоконный кабель.

Для решения проблемы коллизий (когда два или более компьютеров одновременно пытаются передать данные) в сетях с кольцевой топологией применяется **метод маркерного доступа**. Специальное короткое сообщение-маркер постоянно циркулирует по кольцу. Прежде чем передать данные, компьютер должен дождаться маркера, прикрепить данные и служебную информацию к нему и передать это сообщение в сеть.
В быстрых сетях по кольцу циркулируют несколько маркеров.

Существуют две наиболее известных технологии сетей, основанные на кольцевой топологии - *технология Token Ring и технология FDDI*.

Сетевая технология - это согласованный набор стандартных протоколов и реализующих их программно-аппаратных средств, достаточный для построения сети.

В технологии Token Ring реализован метод маркерного доступа, описанный выше.
В технологии FDDI применяется два кольца. При нормальном состоянии сети функционирует только одно из колец, второе позволяет сохранить работоспособность сети в случае отказа узла. Такая сеть обладает высоким быстродействием и чрезвычайной отказоустойчивостью.

Достоинства кольцевой топологии:

- При передачи данных не возникает потери сигнала (благодаря ретрансляции).
- Не возникает коллизий (благодаря маркерному доступу).
- Высокая отказоустойчивость (в технологии FDDI).

Недостатки кольцевой топологии:

- Отказ одного узла может привести к неработоспособности всей сети (в технологии Token Ring).
- Добавление/удаление узла вынуждает разрывать сеть.

Таким образом, кольцевая топология целесообразна для построения надежной или/и высокоскоростной сети, существенное наращивание которой не планируется или маловероятно.

Смешанная топология

Большинство более или менее крупных сетей имеют смешанную топологию, в которой можно выделить отдельные фрагменты типовых топологий (рис. 6).

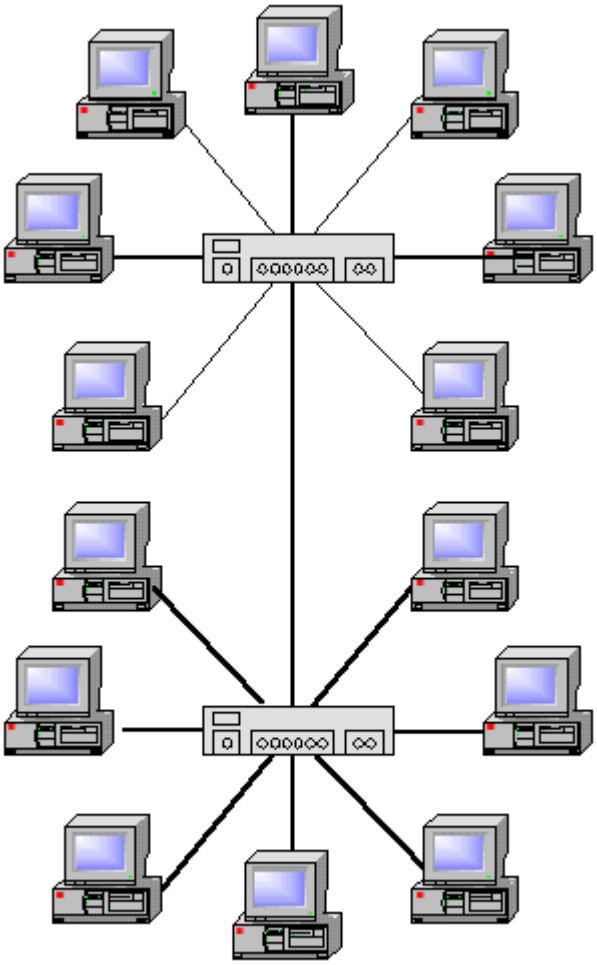


Рис. 6а. Сеть смешанной топологии (звезда-звезда).

Появление смешанных топологий обусловлено, как правило, необходимостью наращивать и модернизировать сеть. Часто суммарные затраты на постепенную модернизацию оказываются существенно большими, а результаты меньшими, чем при тратах на глобальную замену морально устаревших сетей.

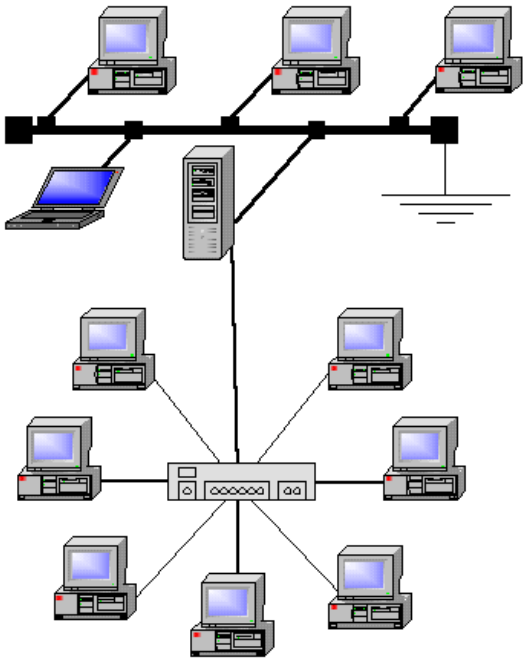


Рис. 6b. Сеть смешанной топологии (звезда-шина).

Сети смешанной топологии обладают достоинствами и недостатками, характерными для составляющих их топологий.

Более подробно о различных топологиях сетей будет рассказано в следующих статьях.

Среда передачи данных

Физическая среда, в которой происходит передача информации, называется **средой передачи данных**.

Можно выделить две основных среды передачи данных (рис. 7):

- проводную (с участием кабелей),
- беспроводную (без участия кабелей).

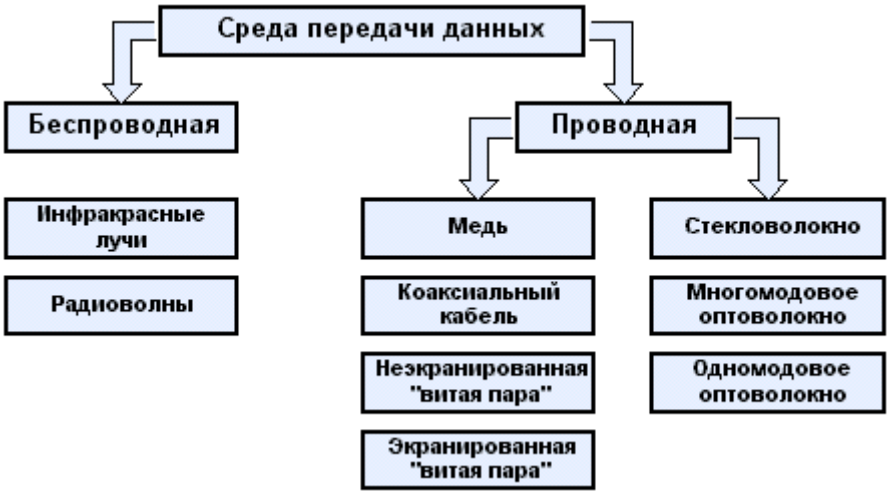


Рис. 7. Среды передачи данных.

К беспроводным средам передачи данных относятся:

- Инфракрасные лучи (соединение компьютеров с помощью инфракрасных портов).
- Радиоволны (передача данных между компьютерами с использованием радиоэфира).

Использование беспроводных сред передачи данных в компьютерных сетях ограничивается несколькими причинами, одна из которых - высокая стоимость. Кроме того, инфракрасная связь действует только в зоне прямой видимости (инфракрасные лучи не могу проникать сквозь стены). На ее основе может быть организована лишь небольшая (часто - временная) сеть внутри одного помещения. Такая сеть, помимо всего прочего, будет работать на довольно низких скоростях. Использование для компьютерной связи радиоволн ограничивается сильной занятостью эфира телевидением, радиовещанием, правительственной, военной и другими видами связи.

Основными **проводными средами передачи данных** являются *медь и стекловолокно*. На их основе изготавливаются различные типы кабелей.

Медную среду передачи данных используют такие типы кабелей как *коаксиальный кабель и «витые пары»* различных категорий.

Коаксиальный кабель в настоящее время для построения новых сетей используются редко. Он обладает низкой пропускной способностью (не более 10 Мбит/с), подвержен действию электромагнитных помех, а сигнал, передаваемый с его помощью, довольно быстро затухает. Все это ограничивает максимально возможную длину сегмента сети до 500 м (при использовании толстого коаксиального кабеля и до 185 м при использовании тонкого коаксиального кабеля), а также максимально возможное число узлов в сети, построенной на основе коаксиального кабеля (до 30 узлов для тонкой Ethernet с использованием коаксиального кабеля RG-8 и RG-11 и до 100 узлов для толстой Ethernet с использованием коаксиального кабеля RG-58). Кроме того, данные, передаваемые посредством коаксиального кабеля, легко перехватить. Однако низкая стоимость коаксиального кабеля и простота монтирования оборудования обуславливают «живучесть» сетей, построенных на его основе.

Кабель «витая пара» получил свое название из-за использования в качестве среды передачи данных одной, двух или четырех пар скрученных медных проводников. Скрученность позволяет гасить помехи, создаваемые каждым из проводников.

Существует две основных разновидности «витой пары» - *неэкранированная (UTP) и экранированная (STP)*. Неэкранированная «витая пара», в свою очередь, подразделяется на несколько категорий. Отличие между UTP и STP в том, что кабель экранированной «витой пары» покрыт защитным экраном - алюминиевой или полиэстеровой оболочкой.

Сети на основе неэкранированной «витой пары» имеют пропускную способность от 1 Мбит/с до 1 Гбит/с (при длине сегмента до 25 м) в зависимости от категории используемого кабеля, максимальную длинную сегмента сети 100 м (сигнал, передаваемый по неэкранированной «витой паре», довольно быстро затухает), рекомендуемое число узлов в сети - 75 (максимально по спецификации - 1024, в реальности - сильно зависит от траффика). Сам кабель сильно подвержен электромагнитным помехам, данные, передаваемые с его помощью, несложно перехватить. Однако UTP имеет невысокую стоимость и легок в прокладке. Вышесказанное обуславливает большую популярность сетей на основе неэкранированной «витой пары».

Сети на основе экранированной «витой пары» имеют более высокую пропускную способность (теоретически: до 500 Мбит/с на расстояние 100 м), максимальную длинную сегмента сети 100 м (сигнал, передаваемый по STP и UTP затухает одинаково быстро), максимальное число узлов по спецификации - 270 (сильно зависит от траффика), а за счет наличия экрана такие сети в значительно меньшей степени подвержены электромагнитным помехам. Данные, передаваемые посредством экранированной «витой пары» перехватить сложнее. В тоже время экранированная «витая пара» имеет большую стоимость и более трудную прокладку, чем неэкранированная.

На основе **стекловолокна** изготавливаются *многомодовые и одномодовые волоконно-оптические кабели*, различающиеся по траектории прохождения световых путей.

В **одномодовом кабеле** все лучи проходят практически один и тот же путь и одновременно достигают приемника.
В **многомодовом кабеле** траектории лучей имеют существенный разброс, что приводит к искажению информации при передаче на большие расстояния.

Соответственно, сети на одномодовых кабелях имеют большую пропускную способность и максимальную длинную сегмента. В то же время они отличаются более высокой стоимостью о сравнению с многомодовыми.

В настоящее время использование оптоволокна становится все более популярным в том числе вследствие снижения его стоимости. Сети, построенные на основе оптоволокна, имеют чрезвычайно высокую пропускную способность (от 100 Мбит/с до 2 Гбит/с и более), не подвержены действию электромагнитных помех, а сигнал, передаваемый по оптоволокну, имеет низкое затухание, что позволяет прокладывать его на значительные расстояния, измеряемые километрами. Оптоволокно не дает утечки сигнала, что делает его надежным в плане перехвата информации. Вместе с тем, как сам кабель, так и оборудование к нему и работы по его прокладке отличаются существенно большей стоимостью по сравнению с медными средами передачи данных. Кабель также подвержен влиянию различных климатических условий

2. Понятие о кластеризации: основные определения и термины. Классификация. Сферы применения.

Кластеризация – это распределение аппаратуры и программного обеспечения по узлам, которые работают вместе как единая система с тем, чтобы гарантировать продолжение функционирования пользовательских приложений во время чрезмерных нагрузок, либо в случае выхода из строя одного из узлов кластера.

Кластеризация становится все более популярной, благодаря недавним улучшениям программного обеспечения управления внешней памятью и приложений, что облегчает этот процесс и делает его более приемлемым в ценовом отношении. А это особенно важно сейчас, когда руководители организаций скупы в отношении расходов на ИТ. Конечно, наличие мощных и очень надежных серверов очень привлекательно, но они весьма дороги. Поэтому многие компании, включая Oracle, используют недорогие массово выпускаемые серверы. Но такой подход ведет к тому, что в сравнении с мощным сервером нагрузка на каждый “малый” сервер меньше и вероятность его сбоя выше.

Ключевым становится следующий вопрос: "Как обеспечить необходимую мощность и надежность уровня предприятия нашим приложениям при условии применения менее надежных серверов?" Ответ: построение эффективной кластеризации.

Однако сразу же возникает множество проблем, как, например, соответствие потребностям приложений, состав необходимой аппаратуры, структура программного обеспечения. Возможность кластеризации может быть учтена уже при проектировании программного обеспечения приложений. Мы рассмотрим различные способы создания кластеров и покажем, что построение эффективных кластеров не сводится к применению одного единственного подхода, необходимо рассмотрение ряда возможностей, из которых для реализации отбираются наиболее походящие для ваших приложений.

Аппаратная кластеризация

Кластеры можно разделить по категориям в соответствии с назначением их основного использования:

- Кластеры высокой готовности (high-availability clusters), или отказоустойчивые (failover), применяются для того, чтобы не допустить прекращения обслуживания в случае выхода из строя основного сервера. Как правило, в этом случае используется дублирующий сервер.
- Кластеры с балансировкой нагрузки (load-balancing clusters) обеспечивают более эффективное использование ресурсов вычислительной системы. В случае высоких нагрузок на серверы, запросы перенаправляются на наименее загруженные серверы.
- Кластеры высокой производительности (high-performancesters clusters) обычно применяются для достижения высокой скорости вычислений. Типичные для этого случая приложения: прогнозирование, в том числе погоды, и научные вычисления. Для получения результатов за короткий промежуток необходимо параллелизировать вычисления. С этой целью первыми были использованы кластерные системы с массивно-параллельной обработкой данных (massively parallel processing (MPP)).

На уровне аппаратуры можно и далее продолжить классификацию: кластер из ПК, кластер из рабочих станций, кластер из SMP-серверов (SMP - symmetric multiprocessing - многопроцессорной симметричной архитектуры) с операционными системами Linux, Solaris, NT и т.д. Очень важен правильный выбор аппаратуры, которая наилучшим образом соответствует вашим потребностям, и способа соединения серверов. Ряд технологий высокопроизводительной коммуникации пакетов и переключения могут быть использованы для соединения рабочих станций, ПК и серверов, входящих в кластеры. Но вместо этих технологий вы можете отдать предпочтение Ethernet, это зависит от производительности и уровня высокой готовности (high-availability) вашей вычислительной среды. Для повышения пропускной

способности сети архитекторы/проектировщики могут выбирать между 100 МБ сетевыми картами и Gigabit Ethernet для получения нужной скорости передачи данных. Другие варианты – это Myrinet, SCI, FC-AL, Giganet, GigE и ATM, но в каждом из этих случае цена кусается.

Очень важно скомпоновать аппаратуру кластера наилучшим образом. Более сложно применение разнородных кластеров, аппаратура которых относится к различным архитектурам, так как узлы могут подсоединяться (и отключаться) к кластеру в разные моменты времени.

Крайне желательна в аппаратных кластерах внешняя память со средствами зеркалирования (mirroring storage) для защиты от сбоев среды хранения данных. Например, в случае простого двух узлового кластера совместно используемая внешняя память может состоять из диска с двумя портами (dual-ported disk), к которому можно обращаться обоих узлов. В этом случае также могут быть нужны специальные кабели для соединения сетевых карт/коммутаторов/хабов.

Если для защиты кластера от системного сбоя применяется “холодное” резервирование ("cold" standby), то необходимо ручное переключение от засбоившего основного сервера к запасному. Но такой подход приводит к прерыванию работы приложения на некоторое время, так как запасной сервер нужно запустить, а приложение перестартовать. “Горячее” резервирование включает автоматическое переключение с сбоившего основного сервера на запасной, который до этого не выполнял работы. В этом случае запасной сервер запускается автоматически и “перехватывает” нагрузку с основного.

Но ни один из этих двух (холодное, горячее) способов резервирования не исправляет повреждения корневой файловой системы (root file systems). Для разрешения этой проблемы в аппаратных кластерах иногда используют свои собственные загрузчики (boot drive), либо средства зеркалирования, реализованные на уровне аппаратуры.

Так как приложения часто становятся недоступными в результате сбоев дисков, то системные администраторы, как правило, используют запасные серверы со средствами зеркалирования дисков, а также технологию RAID (Redundant Arrays of Independent Disks). В этом случае информация хранится на нескольких дисках, которые являются зеркалами друг друга.
Важно понимать особенности различных архитектур аппаратных кластеров:

1. В архитектуре с совместно используемой оперативной памятью (shared-memory architecture) множество процессоров используют общую шину памяти, такие системы определяются как SMP-системы. В этой архитектуре пропускная способность шины часто становится проблемой по мере добавления узлов.
2. В архитектуре с совместно используемыми дисками (shared-disk architecture) множество SMP-серверов совместно используют дисковую память для повышения уровня готовности.
3. Архитектура без разделения ресурсов (shared-nothing architecture) предполагает, что у каждого узла своя собственная оперативная память, свои диски и процессоры. Преимуществом такого подхода является задействование большей пропускной способности по мере добавления узлов.

С течением времени аппаратные кластеры постепенно развиваются. В 80-х годах использовались векторные системы. Затем наступила эра суперкомпьютеров и MPP-систем, а теперь используются кластеры и сети распределенных вычислений (grids). Ряд поставщиков предлагают конкурирующие между собой платформы с различными уровнями поддержки для перечисленных выше архитектур аппаратных кластеров, а также параллелизма, чтобы сделать аппаратные кластеры реальностью. Но по прежнему создание больших аппаратных кластеров с N узлами требует тщательного продумывания и планирования, и немалого бюджета.

Кластеры с балансировкой нагрузки

Технологии аппаратной кластеризации называются стратегиями “пещерного человека” ("caveman"), потому что они не очень развиты, наиболее легки в применении и дешевы.

Если одна их них используется как единственное кластеризированное решение, то оно применимо только для простейших приложений типа презентаций. “Пещерная” кластеризация предполагает использование множества узлов (серверов с идентичными установками вашего приложения) плюс некая дополнительная аппаратура на серверных узлах для управления и распределения нагрузки.

Например, наиболее простая “пещерная” технология – это циклическая (round-robin) служба доменных имен DNS (domain name service), которая использует маршрутизатор и DNS-сервер для циклической рассылки различных пользовательских запросов по всем серверам приложений, так что ни один узел не отягощается Так как у каждого узла есть свой IP-адрес, то легко ввести последовательные URL справочные DNS-файлы и связать их с адресами всех узлов: www1.companyXYZ.com с 143.10.25.1, www2.company.com с 143.10.25.2 и т.д.

Маршрутизатор затем распределит пользовательские запросы по списку URL циклическим образом. Масштабирование в этом случае реализуется достаточно легко: чтобы добавить новый сервер, просто дайте ему последовательный URL в справочном файле DNS-сервера и затем прикрепите этот URL к IP-адресу нового сервера.

Следующим шагом за циклической DNS было применение IP-распределителя (sprayer), устройства подобного маршрутизатору, которое располагается между входящими (inbound) пользовательскими запросами и узлами серверов приложений. Этот метод похож на циклическое решение, за исключением того, что IP-распределители “разбрасывают” или перенаправляют запросы к нескольким узлам. IP-распределители более динамичны и менее произвольны в выборе, чем циклические маршрутизаторы, так что недозагруженнные серверы могут быть использованы более эффективно. Однако, IP-распределители требуют применения SSL-декодеров в случае использования протокола SSL (Secure Sockets Layer - протокол защищенных сокетов, гарантирующий безопасную передачу данных по сети).

Еще одной аппаратной альтернативой является применение реверсивных прокси (reverse-proxy) HTTP-серверов, которые используются в основном как защита от атак злоумышленников, но могут применяться и для балансировки нагрузки. Это способ требует использования кэширования, как правило, связываемого с доступом к Web-страниц в оперативной памяти HTTP-серверов, чтобы снять нагрузку, насколько это возможно, с узлов серверов приложений. Реверс-прокси метод требует использования циклической кластеризации, чтобы предохранить HTTP-серверы от перегрузки.

Другие способы балансировки нагрузки – это единый IP-образ на стороне сервера (server-side single IP image) и трансляция сетевых адресов (network-address translation); оба эти способа дороже и сложнее и требуют изменений заголовков пакетов на основе особенностей нагрузки.

Проблемы отказоустойчивости кластеров с балансировкой нагрузки

Основная проблема по части аппаратуры в разрешении вопросов кластеризации заключается в том, что никакая аппаратура не позволяет удовлетворительно справиться с сбоями узлов. Если компонент сервера (или приложения) засбоил в “пещерной” системе, то весьма вероятно, что пользователь подумает, что вышла из строя вся система. Например, предположим, что внезапно засбоил четвертый узел в циклической или реверс-прокси установке. Если этот узел включен список циклического опроса, то любой входящий пользователь, скорее всего, получит сообщение об ошибке DNS, так как этот сервер не сможет ответить. Чтобы продолжить работу, пользователь должен выйти из системы и в новом сеансе запустить маршрутизатор.

Кроме того, именно аппаратура балансировки нагрузки (DNS-сервер или IP-распределитель) может стать “узким местом” и, тем самым, той единой точкой отказа, выход из строя которой обрушивает всю систему. Поэтому никакая “пещерная” технология не подходит для таких приложений, в которых, например, пользовательские данные должны

быть внесены экран за экраном (как в карточных приложениях обслуживания покупок). В случае же “пещерной” установки при выходе из строя узла, обслуживающего активного пользователя, теряется вся информация данной сессии.

Для “пещерных” систем характерны высокие цены. Технология RAID требует много устройств для работы с дисками, и эти расходы быстро растут по мере масштабирования как ваших приложений, так и самого кластера. Время простоя пользовательских приложений может слишком дорого стоить, особенно для жизненно-важных приложений.

Итак, что касается технологий аппаратной кластеризации, самое главное заключается в том, что для них возможна только почти тотальная защита (но она очень дорога), и такие решения требуют наличия администраторов, которые умеют конфигурировать подобные системы, справляться со сбоями дисков, соединять компоненты кластеров, а также решать сложные сетевые проблемы.

Программные решения: Web-кэшированная кластеризация

Возможно применить программное обеспечение для того, чтобы разрешить то, что по существу является аппаратной проблемой. Отказоустойчивые возможности такой программной инфраструктуры, какой, например, является Oracle9i Application Server (Oracle9iAS), могут обеспечить и динамическую балансировку нагрузки, и высокую готовность, необходимую для сложных и критичных приложений, и стоить только часть цены аппаратного кластера.

Создание кластеров на базе Oracle9iAS Web Cache – это замечательный пример использования программного обеспечения для преодоления сбоев и управления трафиком приложений. Web-кэш предшествует узлам-серверам и подобно обычному кэшу отвечает на все входящие HTTP-запросы и распределяет эти запросы согласно возможностям каждого Web-сервера. В гипотетическом кластере, состоящем из серверов A, B и C, мы сможем сконфигурировать Web Cache для распределения 30% всей нагрузки к Web-серверу A, других 30% к Web-серверу B и 40% к Web-серверу C.

Подобно технологии реверс-прокси сервера, данное решение обладает тем же ключевым преимуществом: если один из этих трех серверов выйдет из строя, Oracle9iAS Web Cache сможет автоматически перераспределять 50% нагрузки по двум остающимся в строю Web-серверам. Когда же засбоивший сервер вернется в строй, Web Cache вновь перераспределит нагрузку по всем трем серверам, и все это будет незаметно, прозрачно для пользователя.

Oracle9iAS Web Cache также поддерживает состояние сессий, не обременяя узлы серверов приложений. Он также обслуживает сайты, которые используют идентификаторы сессии (session ID) и жетоны (cookies). Но поддержка состояния сессий на уровне Web-сервера может быть обременительна, и лучшим решением будет обеспечение минимального, насколько это возможно, набора параметров состояния сессий и только на очень короткие промежутки времени. Для более долгих сессий и больших наборов подобных параметров стоит рассмотреть применение базы данных.

И еще один довод за создание кластеров на базе Oracle9iAS Web Cache – это возможность одного Web Cache взаимодействовать с другими кэшами на этом кластере, чтобы тем самым увеличить общую пропускную способность. Каждый Web Cache обнаруживает новый контент у своего “напарника” и может сохранить его в своем собственном кэше. Также отслеживается и случай выхода из строя “напарника”. Например, если кэш в одном центре данных засбоил, то другие кэши этого кластера могут взять на себя дополнительную нагрузку.

Oracle9iAS Web Cache может не только прозрачно справляться со сбоями узлов, но и также прозрачно управляться со своими собственными сбоями. Вот это по настоящему хорошо!

Кластеризация на уровне J2EE

Oracle9iAS позволяет осуществить кластеризацию на отдельных уровнях J2EE (Java 2 Platform Enterprise Edition): клиентском, Web, EJB (Enterprise JavaBeans) и EIS (Enterpise Information System)—при условии, что приложение спроектировано и разработано в соответствии с четко определенными уровнями. Поэтому, например, приложения с бизнес-логикой на уровне EJB, реализованное с применением уровня JSP (Java Service Pages) не подходит для кластеризации.

Архитекторы/проектировщики всегда должны рассматривать возможность кластеризации на стадии проектирования своих J2EE-приложений. Расщепление уровня J2EE на отдельные уровни позже позволит и далее кластеризировать приложение, обеспечивая тем самым и более высокий уровень высокой готовности в случае сбоев. В недавнем онлайновом опросе об обеспечении максимально возможной высокой готовности J2EE-приложений, подавляющее большинство респондентов отметили, что они рекомендуют разработку приложений с Servlet и EJB на двух уровнях.

Компоненты J2EE-кластеризации

Рассмотренные ранее способы кластеризации были сфокусированы на аспектах масштабирования и производительности кластеров, что само по себе очень важно. Но эти способы не решают жизненно важные проблемы отказоустойчивости (fault-tolerance) приложений, которые обрабатывают большие объемы пользовательских данных за время длительных сессий, иначе говоря, приложений с долго живущими сессиями (long-life session-state applications).

Представьте систему онлайновой торговли, которая требует от пользователей ввести их имена, информацию о счетах, об акциях, которые они хотят купить, и число акций для каждого заказа. И вдруг, когда нажимается кнопка Submit, пользователь получает сообщение об ошибке, так как какая-то ошибка вызвала сбой EJB. Повторный ввод всех этих данных - и потеря денег, так как приложение не может воспроизвести состояние сессии, и возможная потеря пользователя.

Для решения этой критической проблемы для приложений, обремененных обширными данными состояния, Oracle9iAS Containers for J2EE поддерживают "cluster islands" (кластерные острова), наборы серверов на уровне J2EE, на котором параметры состояния сессии могут быть значительно легче воспроизведены, обеспечивая, тем самым, прозрачное перенаправление запроса клиента к другому компоненту, который сможет обслужить этот запрос, если некоторый J2EE-компонент выйдет из строя.

Как правило, проблема поддержки параметров состояния ведет к снижению производительности, независимо от того, находятся ли они в оперативной памяти или параметры состояния сессии хранились в базе данных (в этом случае снижение производительности является результатом выполнения операций ввода-вывода с внешними устройствами). Но поскольку “кластерные острова” (cluster islands) обеспечивают отказоустойчивость на уровне компонентов, параметры состояния могут быть воспроизведены и обеспечены на уровне J2EE без снижения производительности.

Принятие решений

Располагая всеми этими возможностями кластеризации, архитекторы приложений способны принимать обоснованные бизнес-решения. Крайне важно различать способы кластеризации, опирающиеся на аппаратуру, и на программное обеспечение (включая аспекты сетевой инфраструктуры и инфраструктуры хранения данных, которые не были рассмотрены в этой статье).

Не менее важно определить расходы, связанные выходом приложений из строя. Ответственные приложения, такие как онлайновая торговля или обработка записей о пациентах госпиталей, в случае сбоев могут вызвать большие потери. С другой стороны, простые, презентационного типа приложения могут хорошо обслуживаться простыми “пещерными” технологиями.

Как архитектор приложений, вы должны рассмотреть ряд вопросов, связанных с кластеризацией: Что произойдет, если приложение выйдет из строя из-за какой-то своей ошибки или отключения питания или здание, в котором расположено ваше оборудование, сгорит? Что если откажет важный электронный компонент, или испортится корневая файловая система, приведя к краху резервные (standby) машины, или на вашем основном диске появились сбойные секторы? В какой защите нуждается ваше приложение, и как много вы готовы заплатить за такую защиту (или за ее отсутствие)?

Вопросы возможности кластеризации должны играть важную роль во всем процессе разработки приложения. В начале проекта проведите встречу со всеми заинтересованные стороны, включая руководителей уровня С, чтобы определить критичность приложения и, соответственно, необходимость применения кластеризации. Привлекайте в ваши дискуссии сетевых и баз данных администраторов, чтобы определить инфраструктуру, которая можно поддержать в рамках заданных ограничений по персоналу и бюджету.

И, наконец, приложения должны быть спроектированы и разработаны с использованием четко определенных уровней - клиент, Web-, EJB- и EIS- уровни. (Это очень хорошее правило в любых обстоятельствах.) По мере развития приложения его потребности в кластеризации, вероятно, будут изменяться, так что старайтесь сохранять гибкость своих приложений, насколько это возможно.

Сквозная кластеризация

Oracle9iAS предоставляет архитекторам приложений широкий набор готовых к развертыванию средств, которые помогут при решении в случаях сложной неочевидной кластеризации. Способы кластеризации простираются от простых “пещерных” методов для балансировки нагрузки до методов кэш-кластеризации приложений с богатых информационным содержанием. Кластеризация может быть организована на уровне Web-сервера и на J2EE-уровне. Архитекторы приложений могут даже выбрать кластеризацию на уровне отдельных компонентов J2EE, а с помощью “кластерных островов” (cluster islands) они могут поддерживать состояние сессий без потери производительности. Oracle9iAS – это действительно полный, всесторонний (end-to-end) инструмент реализации кластеризованных решений, способных противостоять сбоям на любых уровнях методами, прозрачными для конечных пользователей.

В данной статье вопросы кластеризации были только затронуты. Более детальные советы и технические рекомендации можно найти в “Белой Книге” о кластеризации на Oracle9iAS и на бесплатном Internet-семинаре по J2EE-кластеризации. Оба источника, как и многие другие ресурсы, доступны на OTN в Middleware Architecture Series.

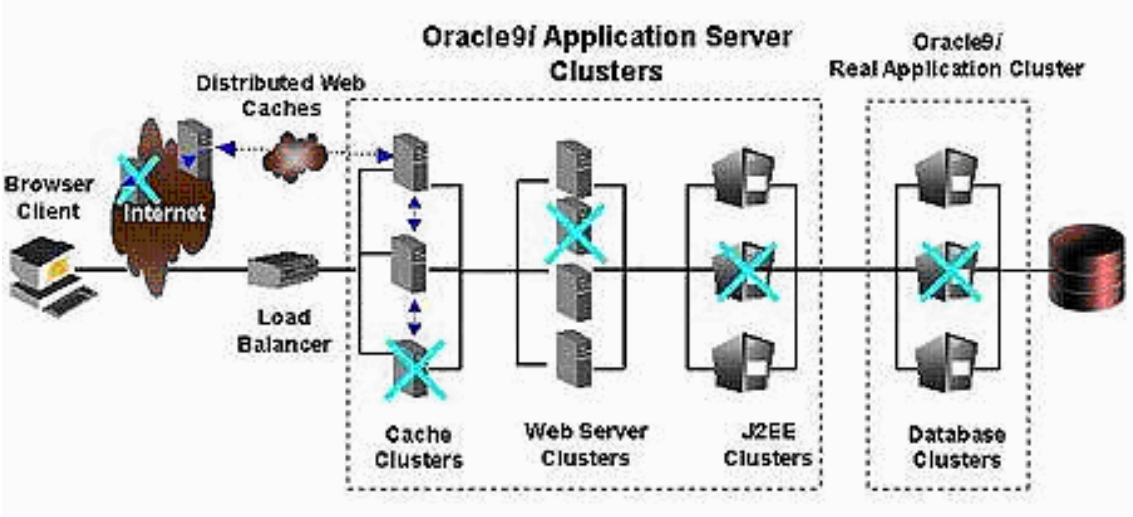


Рисунок 1 Сквозная кластеризация

Кластеризация на базе Oracle9iAS позволяет архитекторам приложений защититься от серверных сбоев, независимо от того, где эти сбои имели место.

3. Модель OSI.

Модель OSI определяет уровни взаимодействия систем в сетях с коммутацией пакета, стандартные названия уровней, функции, которые должен выполнять каждый уровень.

Средства взаимодействия делятся на 7 уровней : прикладной, представительный, сеансовый, транспортный, сетевой, канальный и физический.

В модели OSI есть два типа протоколов- с установленным соединением (телефон) и без предварительной установки соединения (почтовый ящик).

Физический уровень- передача потока битов по кабелям. Не вникает в смысл информации, которую передаёт.

Канальный уровень- проверка доступности среды передачи и механизм обнаружения и коррекции ошибок. Формирует по определённому алгоритму контрольную сумму. Протоколы канального уровня реализуются компьютерами, мостами, маршрутизаторами. В компах функции канального уровня реализуются совместными усилителями сетевых адаптеров и их драйверов.

Сетевой уровень служит для образования единой транспортной системы, объединяющей несколько сетей. Согласование разных технологий, упрощение адресации в крупных сетях и создание надёжных и гибких барьеров на пути нежелательного трафика между сетями. Сообщения сетевого уровня- пакеты. Маршрутизаторы.

Транспортный уровень обеспечивает приложениям или верхним уровням стека- прикладному и сеансовому- передачу данных с той степенью надёжностью, которая им требуется. Модель OSI определяет 5 видов сервиса , предоставляемых транспортным уровнем- эти виды сервисов отличаются качеством предоставляемых услуг- срочностью, возможностью восстановления прерванной связи, возможностью к обнаружения и исправлению ошибок передачи, таких как искажение, потеря и дублирования пакета.

Сеансовый уровень фиксирует, какая из сторон является активной в настоящее время, предоставляя средства синхронизации.

Представительный уровень – на этом уровне может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данных обеспечивается сразу для всех прикладных служб.

Прикладной уровень – набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам (файлам, принтерам, организуется совместная работа например при помощи протокола электронной службы).

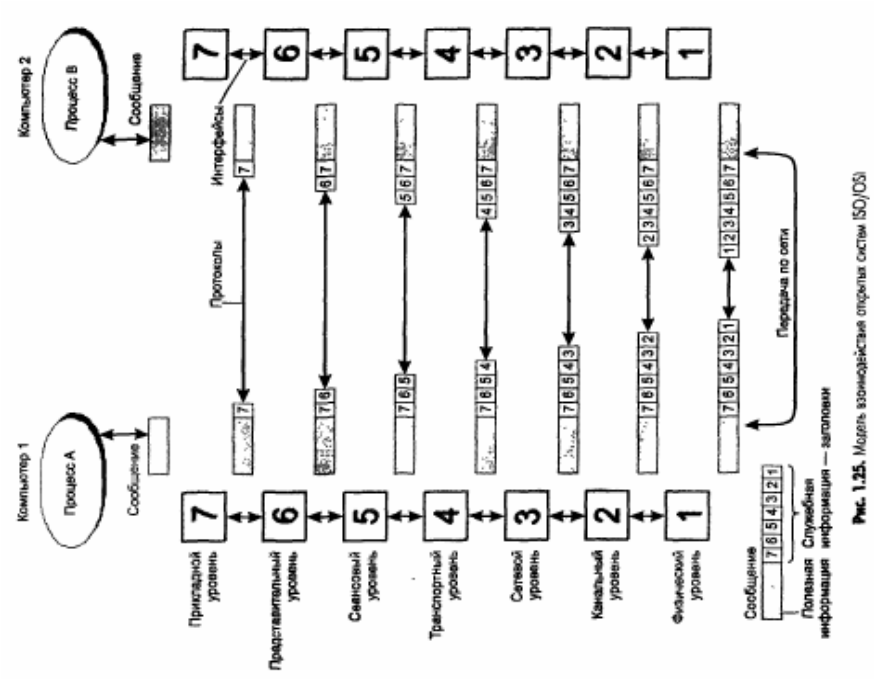


Рис. 1.25. Модель взаимодействия открытых систем ISO/OSI

4. Архитектура систем хранения данных SAN

Сеть Хранения Данных (SAN) или Система Хранения Данных (далее СХД) – это специализированное железо и ПО, предназначенное для работы с огромными массивами ценной информации.

Основные проблемы, решаемые СХД

Рассмотрим типичные проблемы, связанные с растущими объёмами информации в любой организации. Предположим, что это хотя бы несколько десятков компьютеров и несколько разнесённых территориально офисов.

1. Децентрализация информации – если раньше все данные могли храниться буквально на одном жёстком диске, то сейчас любая функциональная система требует отдельного хранилища – к примеру, серверов электронной почты, СУБД, домена и так далее. Ситуация усложняется в случае распределённых офисов (филиалов).
2. Лавинообразный рост информации – зачастую количество жёстких дисков, которые вы можете установить в конкретный сервер, не может покрыть необходимую системе ёмкость.

Как следствия:

Как следствие, невозможность полноценно защитить хранимые данные – действительно, ведь довольно трудно произвести даже backup данных, которые находятся не только на разных серверах, но и разнесены территориально.

Недостаточная скорость обработки информации – каналы связи между удалёнными площадками пока оставляют желать лучшего, но даже при достаточно «толстом» канале не всегда возможно полноценное использование существующих сетей, например, IP, для работы.

Сложность резервного копирования (архивирования) – если данные читаются и записываются небольшими блоками, то произвести полное архивирование информации с удалённого сервера по существующим каналам может быть нереально – необходима передача всего объёма данных. Архивирование на местах зачастую нецелесообразно по финансовым соображениям – необходимы системы для резервного копирования (ленточные накопители, например), спе-

циальное ПО (которое может стоить немалых денег), обученный и квалифицированный персонал.

3. Сложно или невозможно предугадать требуемый объём дискового пространства при развертывании компьютерной системы.

Как следствия:

Возникают проблемы расширения дисковых ёмкостей – довольно сложно получить в сервере ёмкости порядков терабайт, особенно если система уже работает на существующих дисках небольшой ёмкости – как минимум, требуется остановка системы и неэффективные финансовые вложения.

Неэффективная утилизация ресурсов – порой не угадать, в каком сервере данные будут расти быстрее. В сервере электронной почты может быть свободен критически малый объём дискового пространства, в то время как другое подразделение будет использовать всего лишь 20% объёма недешёвой дисковой подсистемы (например, SCSI).

4. Низкая степень конфиденциальности распределённых данных – невозможно проконтролировать и ограничить доступ в соответствии с политикой безопасности предприятия. Это касается как доступа к данным по существующим для этого каналам (локальная сеть), так и физического доступа к носителям – к примеру, не исключены хищения жёстких дисков, их разрушение (с целью затруднить бизнес организации). Неквалифицированные действия пользователей и обслуживающего персонала могут нанести ещё больший вред. Когда компания в каждом офисе вынуждена решать мелкие локальные проблемы безопасности, это не даёт желаемого результата.
5. Сложность управления распределёнными потоками информации – любые действия, которые направлены на изменения данных в каждом филиале, содержащем часть распределённых данных, создает определённые проблемы, начиная от сложности синхронизации различных баз данных, версий файлов разработчиков и заканчивая ненужным дублированием информации.
6. Низкий экономический эффект внедрения «классических» решений – по мере роста информационной сети, больших объёмов данных и всё более распределённой структуры предприятия финансовые вложения оказываются не столь эффективны и зачастую не могут решить возникающих проблем.
7. Высокие затраты используемых ресурсов для поддержания работоспособности всей информационной системы предприятия – начиная от необходимости содержать большой штат квалифицированного персонала и заканчивая многочисленными недешёвыми аппаратными решениями, которые призваны решить проблему объёмов и скоростей доступа к информации вкупе с надёжностью хранения и защитой от сбоев.

Рассмотрим типовые схемы подключения и виды систем хранения данных.

Уровни защиты

RAID

В основе всех систем хранения данных лежит практика защиты информации на базе технологии RAID – без этого любая технически продвинутая СХД будет бесполезна, потому что жёсткие диски в этой системе являются самым ненадёжным компонентом. Организация дисков в RAID – это «нижнее звено», первый эшелон защиты информации и повышения скорости обработки. Архитектура RAID на **Рисунок 2.**

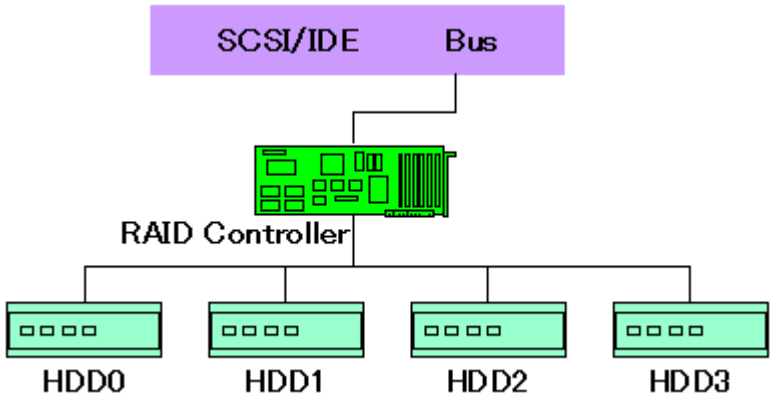


Рисунок 2 Архитектура RAID

При организации RAID в любых системах хранения данных дополнительно к защите информации мы получаем несколько неоспоримых преимуществ, одно из которых – скорость доступа к информации.
С точки зрения пользователя или ПО, скорость определяется не только пропускной способностью системы (Мбайт/с), но и числом транзакций – то есть числом операций ввода-вывода в единицу времени (IOPS). Увеличению IOPS способствует, что вполне логично, большее число дисков и те методики повышения производительности, которые предоставляет контроллер RAID (к примеру, кэширование).
Если для просмотра потокового видео или организации файл-сервера больше важна общая пропускная способность, то для СУБД, любых OLTP (online transaction processing) приложений критично именно число транзакций, которые способна обрабатывать система. А с этим параметром у современных жёстких дисков всё не так радужно, как с растущими объёмами и, частично, скоростями. Все эти проблемы призвана решить сама система хранения данных – ниже будет видно, как и какими методами.

Системы DAS

Устройства **DAS (Direct Attached Storage)** – системы хранения, подключаемые напрямую к серверу. Сюда относятся как самые простые SCSI-системы, подключаемые к SCSI/RAID-контроллеру сервера, так и устройства FibreChannel, подключенные прямо к серверу, хотя и предназначены они для сетей SAN. В этом случае топология DAS является вырожденной SAN (сетью хранения данных) (**Рисунок 3**)

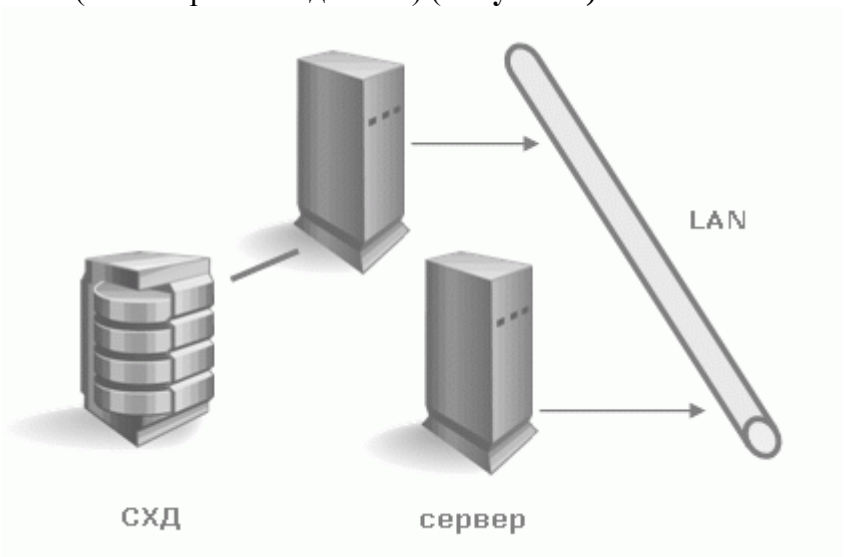


Рисунок 3 Архитектура DAS

В этой схеме один из серверов имеет доступ к данным, хранящимся на СХД. Клиенты получают доступ к данным, обращаясь к этому серверу через сеть. То есть сервер имеет блочный доступ к данным на СХД, а уже клиенты пользуются файловым доступом – эта концепция очень важна для понимания. Минусы такой топологии очевидны:

1. Низкая надежность – при проблемах сети или аварии сервера данные становятся недоступны всем сразу.
2. Высокая латентность, обусловленная обработкой всех запросов одним сервером и использующимся транспортом (чаще всего – IP).
3. Высокая загрузка сети, часто определяющая пределы масштабируемости путём добавления клиентов.
4. Плохая управляемость – вся ёмкость доступна одному серверу, что снижает гибкость распределения данных.
5. Низкая утилизация ресурсов – трудно предсказать требуемые объёмы данных, у одних устройств DAS в организации может быть избыток ёмкости (дисков), у других её может не хватать – перераспределение часто невозможно или трудоёмко.

Системы NAS

Устройства **NAS (Network Attached Storage)** – устройства хранения, подключённые напрямую в сеть. В отличие от других систем NAS обеспечивает файловый доступ к данным и никак иначе. NAS-устройства представляют из себя комбинацию системы хранения данных и сервера, к которому она подключена. В простейшем варианте обычный сетевой сервер, предоставляющий файловые ресурсы, является устройством NAS (**Рисунок 4**)

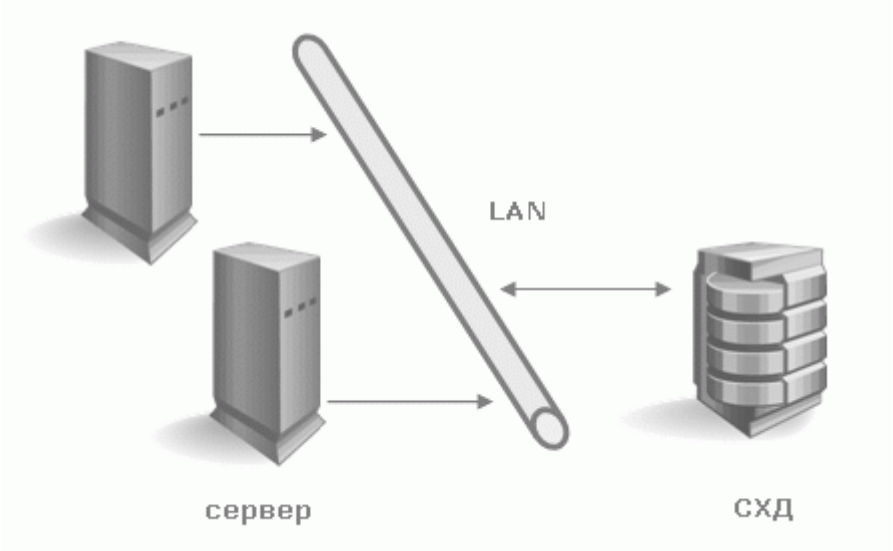


Рисунок 4 Архитектура NAS

Все минусы такой схемы аналогичны DAS-топологии, за некоторым исключением. Из добавившихся минусов отметим возросшую, и часто значительно, стоимость – правда, стоимость пропорциональна функциональности, а тут уже часто «есть за что платить». NAS-устройства могут быть простейшими «коробочками» с одним портом ethernet и двумя жёсткими дисками в RAID1, позволяющими доступ к файлам по лишь одному протоколу CIFS (Common Internet File System) до огромных систем в которых могут быть установлены сотни жёстких дисков, а файловый доступ обеспечивается десятком специализированных серверов внутри NAS-системы. Число внешних Ethernet-портов может достигать многих десятков, а ёмкость хранимых данных – несколько сотен терабайт (например EMC Celerra CNS). Такие модели по надёжности и производительности могут далеко обходить многие midrange-устройства SAN. Что интересно, NAS-устройства могут быть частью SAN-сети и не иметь собственных накопителей, а лишь предоставлять файловый доступ к данным, находящимся на блочных устройствах хранения. В таком случае NAS берёт на себя функцию мощного

специализированного сервера, а SAN – устройства хранения данных, то есть мы получаем топологию DAS, скомпонованную из NAS- и SAN-компонентов.
NAS-устройства очень хороши в гетерогенной среде, где необходим быстрый файловый доступ к данным для многих клиентов одновременно. Также обеспечивается отличная надёжность хранения и гибкость управления системой вкупе с простотой обслуживания. На надёжности особо останавливаться не будем – этот аспект СХД рассмотрен выше. Что касается гетерогенной среды, доступ к файлам в рамках единой NAS-системы может быть получен по протоколам TCP/IP, CIFS, NFS, FTP, TFTP и другим, включая возможность работы NAS, как iSCSI-target, что обеспечивает функционирование с различным ОС, установленными на хостах. Что касается лёгкости обслуживания и гибкости управления, то эти возможности обеспечиваются специализированной ОС, которую трудно вывести из строя и не нужно обслуживать, а также простотой разграничения прав доступа к файлам. К примеру, возможна работа в среде Windows Active Directory с поддержкой требуемой функциональности – это может быть LDAP, Kerberos Authentication, Dynamic DNS, ACLs, назначение квот (quotas), Group Policy Objects и SID-истории. Так как доступ обеспечивается к файлам, а их имена могут содержать символы различных языков, многие NAS обеспечивают поддержку кодировок UTF-8, Unicode. К выбору NAS стоит подходить даже тщательнее, чем к DAS-устройствам, ведь такое оборудование может не поддерживать необходимые вам сервисы, например, Encrypting File Systems (EFS) от Microsoft и IPSec. К слову можно заметить, что NAS распространены намного меньше, чем устройства SAN, но процент таких систем всё же постоянно, хотя и медленно, растёт – в основном за счёт вытеснения DAS.

Системы SAN

Устройства для подключения в **SAN (Storage Area Network)** – устройства для подключения в сеть хранения данных. Сеть хранения данных (SAN) не стОит путать с локальной сетью – это различные сети. Чаще всего SAN основывается на стеке протоколов FibreChannel и в простейшем случае состоит из СХД, коммутаторов и серверов, объединённых оптическими каналами связи. На рисунке мы видим высоконадёжную инфраструктуру, в которой серверы включены одновременно в локальную сеть (слева) и в сеть хранения данных (справа):

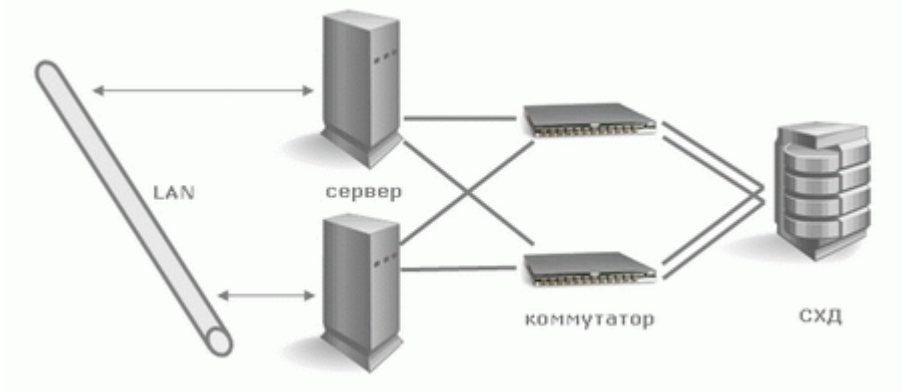


Рисунок 5 Архитектура SAN

После довольно детального рассмотрения устройств и принципов их функционирования нам будет довольно легко понять топологию SAN. На рисунке мы видим единую для всей инфраструктуры СХД, к которой подключены два сервера. Серверы имеют резервированные пути доступа – в каждом установлено по два HBA (или один двухпортовый, что снижает отказоустойчивость). Устройство хранения имеет 4 порта, которыми оно подключено в 2 коммутатора. Предполагая, что внутри имеется два резервируемых процессорных модуля, легко догадаться, что лучшая схема подключения – когда каждый коммутатор подключён и в первый, и во второй процессорный модуль. Такая схема обеспечивает доступ к любым данным, находящимся на СХД, при выходе из строя любого процессорного модуля, коммутатора или пути доступа. Надёжность СХД нами уже изучена, два коммутатора и две фабрики ещё более

увеличивают доступность топологии, так что если из-за сбоя или ошибки администратора один из коммутационных блоков вдруг отказал, второй будет функционировать нормально, ведь эти два устройства не связаны между собой.

Показанное подключение серверов называется подключением с высокой доступностью (high availability), хотя в сервере при необходимости может быть установлено ещё большее число HBA. Физически каждый сервер имеет только два подключения в SAN, однако логически система хранения доступна через четыре пути – каждая HBA предоставляет доступ к двум точкам подключения на СХД, к каждому процессорному модулю раздельно (эту возможность обеспечивает двойное подключение коммутатора к СХД). На данной схеме самое ненадежной устройство – это сервер. Два коммутатора обеспечивают надежность порядка 99,99%, а вот сервер может отказать по разным причинам. Если необходима высоконадёжная работа всей системы, серверы объединяются в кластер, приведённая схема не требует никакого аппаратного дополнения для организации такой работы и считается эталонной схемой организации SAN. Простейший же случай – серверы, подключённые единственным путем через один свитч к системе хранения. Однако система хранения при наличии двух процессорных модулей должна подключаться в коммутатор как минимум одним каналом на каждый модуль – остальные порты могут быть использованы для прямого подключения серверов к СХД, что иногда необходимо. И не стоит забывать, что SAN возможно построить не только на базе FibreChannel, но и на базе протокола iSCSI – при этом можно использовать только стандартные ethernet-устройства для коммутации, что удешевляет систему, но имеет ряд дополнительных минусов (оговоренных в разделе, рассматривающем iSCSI). Также интересна возможность загрузки серверов с системы хранения – не обязательно даже наличие внутренних жёстких дисков в сервере. Таким образом, с серверов окончательно снимается задача хранения каких-либо данных. В теории специализированный сервер может быть превращён в обычную числодробилку без каких-либо накопителей, определяющими блоками которого являются центральные процессоры, память, а так же интерфейсы взаимодействия с внешним миром, например порты Ethernet и FibreChannel. Какое-то подобие таких устройств являют собой современные blade-серверы.
Хочется отметить, что устройства, которые возможно подключить в SAN, не ограничены только дисковыми СХД – это могут быть дисковые библиотеки, ленточные библиотеки (стримеры), устройства для хранения данных на оптических дисках (CD/DVD и прочие) и многие другие.
Из минусов SAN отметим лишь высокую стоимость её компонент, однако плюсы неоспоримы:

1. Высокая надёжность доступа к данным, находящимся на внешних системах хранения. Независимость топологии SAN от используемых СХД и серверов.
2. Централизованное хранение данных (надёжность, безопасность).
3. Удобное централизованное управление коммутацией и данными.
4. Перенос интенсивного трафика ввода-вывода в отдельную сеть, разгружая LAN.
5. Высокое быстродействие и низкая латентность.
6. Масштабируемость и гибкость логической структуры SAN
7. Географически размеры SAN, в отличие от классических DAS, практически не ограничены.
8. Возможность оперативно распределять ресурсы между серверами.
9. Возможность строить отказоустойчивые кластерные решения без дополнительных затрат на базе имеющейся SAN.
10. Простая схема резервного копирования – все данные находятся в одном месте.
11. Наличие дополнительных возможностей и сервисов (снапшоты, удаленная репликация).
12. Высокая степень безопасности SAN.

5. Основные устройства физического уровня модели OSI и их характеристики

Характеристики кабелей-

- **Затухание (Attenuation).** Затухание измеряется в децибелах на метр для определенной частоты или диапазона частот сигнала.
- **Перекрестные наводки на ближнем конце (Near End Cross Talk, NEXT).** Измеряются в децибелах для определенной частоты сигнала.
- **Импеданс (волновое сопротивление)** - это полное (активное и реактивное) сопротивление в электрической цепи. Импеданс измеряется в Омах и является относительно постоянной величиной для кабельных систем (например, для коаксиальных кабелей, используемых в стандартах Ethernet, импеданс кабеля должен составлять 50 Ом). Для неэкранированной витой пары наиболее часто используемые значения импеданса - 100 и 120 Ом. В области высоких частот (100-200 МГц) импеданс зависит от частоты.
- **Активное сопротивление** - это сопротивление постоянному току в электрической цепи. В отличие от импеданса активное сопротивление не зависит от частоты и возрастает с увеличением длины кабеля.
- **Емкость** - это свойство металлических проводников накапливать энергию. Два электрических проводника в кабеле, разделенные диэлектриком, представляют собой конденсатор, способный накапливать заряд. Емкость является нежелательной величиной, поэтому следует стремиться к тому, чтобы она была как можно меньше (иногда применяют термин "паразитная емкость"). Высокое значение емкости в кабеле приводит к искажению сигнала и ограничивает полосу пропускания линии.
- **Уровень внешнего электромагнитного излучения или электрический шум.** Электрический шум - это нежелательное переменное напряжение в проводнике.
- **Диаметр или площадь сечения проводника.** Для медных проводников достаточно употребительной является американская система AWG (American Wire Gauge), которая вводит некоторые условные типы проводников, например 22 AWG, 24 AWG, 26 AWG. Чем больше номер типа проводника, тем меньше его диаметр

Виды кабелей- более подробно на http://www.bmstu.ru/~iu/Vlasov/Pages/Page5_4_1.html.

Виды кабелей:

- Кабели на основе неэкранированной витой пары. Существует 7 категорий этого типа кабеля. Для 1-й категории- требования по скорости минимальны.
- Кабели на основе экранированной витой пары- 9 типов. Для 1-го типа данного типа кабеля волновое сопротивление равно 150 Ом
- Коаксиальные кабели- Толстый коаксиал- волновое сопротивление- 50 Ом, внешний диаметр- 12 мм. Внутренний диаметр 2.17 мм. Затухание на частоте 10 МГц- не хуже 18 дБ/км. Тонкий коаксиал- внешний диаметр- 50 мм., внутренний- 0.89 мм. Волновое сопротивление 50 Ом. Затухание выше, чем в толстом коаксиале, поэтому уменьшают длину.
- Волоконно- оптические кабели-

Основное внимание в современных стандартах уделяется кабелям на основе витой пары и волоконно-оптическим кабелям.

6. Ethernet. Особенности физической реализации.

http://www.bmstu.ru/~iu/Vlasov/Pages/Page8_6.html

Сейчас существуют следующие среды передачи данных:

- **10Base-5** - коаксиальный кабель диаметром центрального медного провода 2,17 мм и внешним диаметром около 10 мм, называемый "толстым" коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента - 500 м (без повторителей). Кабель используется как моноканал для всех станций. Станция должна подключаться к кабелю при помощи приемопередатчика - трансивера (transmitter + receiver = transceiver). Трансивер устанавливается непосредственно на кабеле и питается от сетевого адаптера компьютера. Трансивер может подсоединяться к кабелю как методом прокалывания, обеспечивающим непосредственный физический контакт, так и бесконтактным методом.
- **10Base-2** - коаксиальный кабель диаметром центрального медного провода 0,89 мм и внешним диаметром около 5 мм, называемый "тонким" коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента - 185 м (без повторителей). Станции подключаются к кабелю с помощью высокочастотного (BNC) Т-кон-нектора, который представляет собой тройник, один отвод которого соединяется с сетевым адаптером, а два других - с двумя концами разрыва кабеля. Максимальное количество станций, подключаемых к одному сегменту, - 30. Минимальное расстояние между станциями - 1 м. Кабель "тонкого" коаксиала имеет разметку для подключения узлов с шагом в 1 м. Реализация этого стандарта на практике приводит к наиболее простому решению для кабельной сети, так как для соединения компьютеров требуются только сетевые адаптеры, Т-коннекторы и терминаторы 50 Ом. Однако этот вид кабельных соединений наиболее сильно подвержен авариям и сбоям: кабель более восприимчив к помехам, чем "толстый" коаксиал, в моноканале имеется большое количество механических соединений (каждый Т-коннектор дает три механических соединения, два из которых имеют жизненно важное значение для всей сети), пользователи имеют доступ к разъемам и могут нарушить целостность моноканала.
- **10Base-T** - кабель на основе неэкранированной витой пары (Unshielded Twisted Pair, UTP). Образует звездообразную топологию на основе концентратора. Расстояние между концентратором и конечным узлом - не более 100 м. Конечные узлы соединяются с помощью двух витых пар по топологии "точка-точка" со специальным устройством - многопортовым повторителем. Концентратор осуществляет функции повторителя сигналов на всех отрезках витых пар, подключенных к его портам, так что образуется единая среда передачи данных - логический моноканал.
- **10Base-F** - волоконно-оптический кабель. Функционально сеть Ethernet на оптическом кабеле состоит из тех же элементов, что и сеть стандарта 10Base-T - сетевых адаптеров, многопортового повторителя и отрезков кабеля, соединяющих адаптер с портом повторителя. Как и в случае витой пары, для соединения адаптера с повторителем используется два оптоволокна - одно соединяет выход Tx адаптера с входом Rx повторителя, а другое - вход Rx адаптера с выходом Tx повторителя. Имеется несколько вариантов этой спецификации - FOIRL (расстояние до 1000 м), 10Base-FL (расстояние до 2000 м) (Увеличена мощность передатчиков), 10Base-FB (расстояние до 2000 м) (Повторители при отсутствии кадров для передачи постоянно обмениваются специальными последовательностями сигналов, отличающимися от сигналов кадров данных, для поддержания синхронизации. Поэтому они вносят меньшие задержки при передаче данных из одного сегмента в другой, и это является главной причиной, по которой количество повторителей удалось увеличить до 5.). Число 10 в указанных выше названиях обозначает битовую скорость передачи данных этих стандартов - 10 Мбит/с, а слово "Base" - метод передачи на одной базовой частоте 10 МГц (в отличие от методов, использующих несколько несущих частот, которые называются Broadband - широкополосными). Последний символ в названии стандарта физического уровня обозначает тип кабеля.

7. Протокол FCIP iFCIP.

По мере того как организации сталкиваются с необходимостью хранить, защищать, резервировать и реплицировать огромные объемы данных, они все больше склонны строить свои системы хранения на базе SAN (Storage Area Network). Вместе с тем, чтобы обеспечить всем пользователям доступ к географически распределенным SAN корпорациям, нужны надежные, высокоскоростные и желательно недорогие каналы.

Современные SAN преимущественно базируются на протоколе Fibre Channel (FC) – гигабитовой или мультигигабитовой сетевой технологии, специально разработанной для соединения серверов и удаленных устройств хранения данных (подобно SCSI она реализует последовательную поблочную их передачу). Вопреки своему названию, в качестве среды передачи FC может использовать как оптоволокно, так и медный кабель. При применении одномодового оптоволокна длина канала может превышать 10 км, медные же соединения значительно короче – только около 30 м. Протокол поддерживает широкий спектр скоростей передачи, включая 133, 266, 532 и 10 625 Mbps (эти показатели удваиваются при дуплексном режиме).

Системы SAN получили широкое распространение в корпоративной среде благодаря возможности реализовать в них эффективные методы управления информацией. В то же время для обеспечения непрерывности бизнес-процессов и катастрофоустойчивости возникла необходимость в следующих функциях:

- удаленном архивировании на ленточные носители;
- удаленном зеркалировании дисков;
- разделении на уровне блоков данных, хранящихся на географически распределенных системах;
- разнесении систем хранения на большие расстояния для предотвращения последствий природных или других видов катастроф;
- централизованного управления распределенными ресурсами и ряде других, требующих соединения географически удаленных SAN.

Выбор технологии для объединения SAN зависит от таких факторов, как расстояние, полоса пропускания, стоимость, время ожидания (синхронные или асинхронные приложения) и т. п. Из имеющихся сегодня и появляющихся стандартов организации могут использовать:

- Fibre Channel over DWDM;
- Fibre Channel over SONET;
- Fibre Channel over ATM;
- Fibre Channel over IP.

Fibre Channel over DWDM в состоянии обеспечить бóльшую длину канала, нежели нативный FC, и преимущественно находит применение в сетях масштаба города. Эта технология является идеальной для удаленного зеркалирования, требующего крайне высокой пропускной способности, однако она довольно дорога.

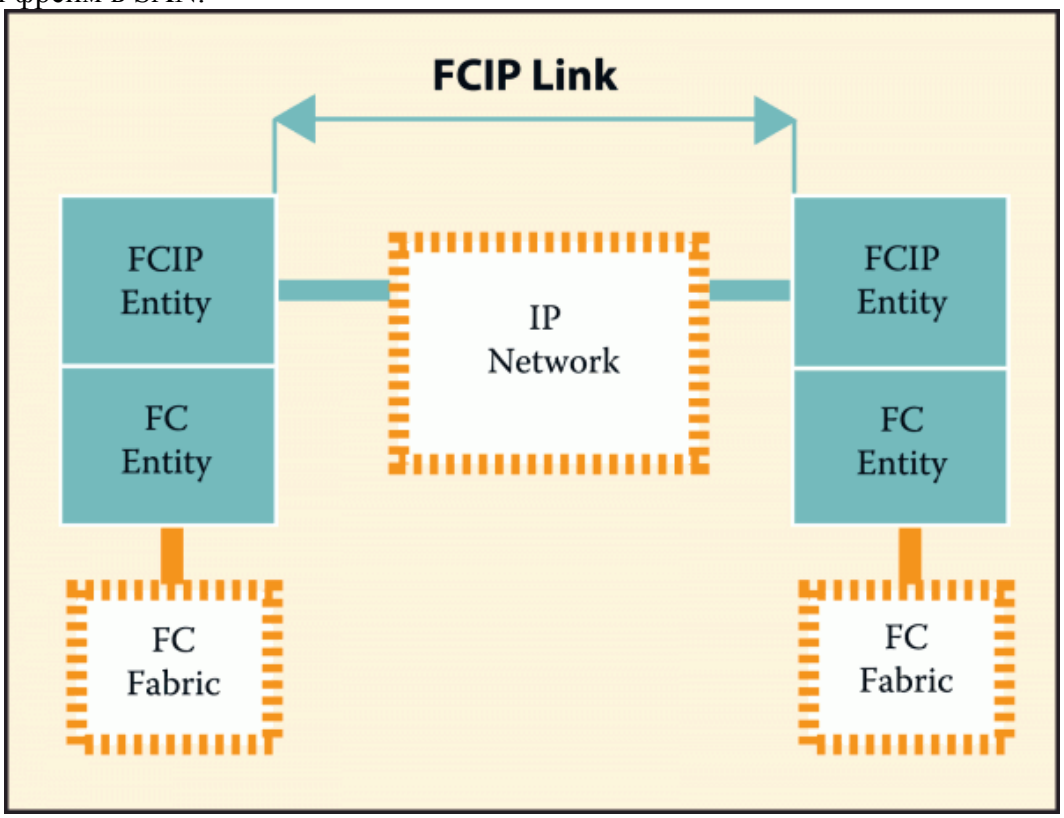
Аналогично ей, если не принимать во внимание методы мультиплексирования, функционирует и Fibre Channel over SONET. При ее использовании сеть логически выглядит как единая система SAN, что делает ее удобной для удаленного зеркалирования.

Fibre Channel over ATM инкапсулирует FC-данные в ATM-ячейки и передает трафик со всеми присущими технологии особенностями, такими как различные уровни гарантированного сервиса и изменяемая полоса пропускания. При соответствующем выборе Class of Service (CoS) данное решение вполне приемлемо как для синхронных, т. е. чувствительных к задержке приложений (например, зеркалирования), так и для асинхронных (резервирования на магнитную ленту).

Fibre Channel over IP (FCIP) рассматривается как идеальная комбинация технологий, позволяющих решить проблемы, возникающие при объединении географически удаленных SAN. С одной стороны, FC – это зрелая технология для построения SAN в масштабах кампуса, в которую корпорациями вложено немало средств и для которой имеется много совмес-

тимых приложений, с другой, – IP-сети являются сегодня наиболее распространенными и наилучшим образом приспособлены для передачи данных через глобальные сети.

Спецификация FCIP разработана группой Internet Engineering Task Force (IETF) и описывает механизм создания прозрачного тоннеля для транспорта FC-фреймов через IP-сети. Выполняемые протоколом FCIP операции во многом подобны любому из тоннельных механизмов. Имеются два оконечных устройства (шлюза), служащие интерфейсом между локальной SAN и IP-сетью. В режиме передачи каждое из них принимает FC-фрейм от SAN и инкапсулирует его в IP-пакет, который затем передается через IP-сеть с использованием TCP в качестве транспортного протокола. Эти же устройства, работающие в режиме приема, получают входящий FCIP-трафик, отбрасывают заголовки IP-пакетов и направляют первоначальный фрейм в SAN.

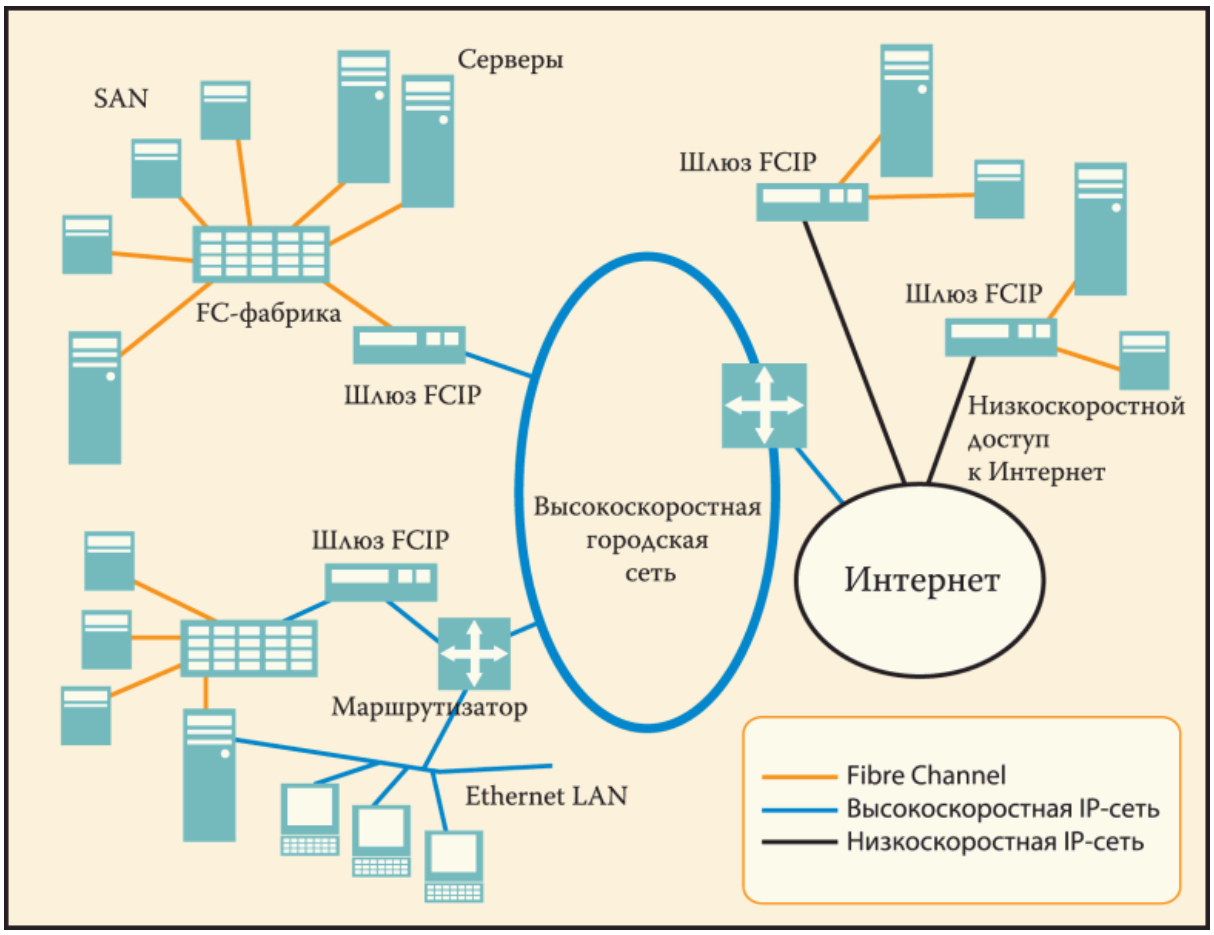


На рис. 1 приведена схема модели тоннеля между двумя SAN. Как видно из рисунка, шлюз между коммутационной фабрикой FC и IP-сетью содержит два логических модуля, которые называются FC Entity (FC сущность) и FCIP Entity (напомним, что в многоуровневой модели OSI сущностью уровня является доступ к нижележащему уровню и предоставление услуг вышележащему).

FC Entity представляет собой специфический функциональный компонент, образующий в комбинации с FCIP Entity интерфейс между коммутационной фабрикой FC и IP-сетью. В свою очередь, FCIP Entity отвечает за обмен пакетами в IP-сети.

FCIP Link (канал FCIP) является базовым сервисом протокола FCIP. Он соединяет две FC-фабрики, используя IP-сеть в качестве транспорта, чтобы образовать единую коммутационную фабрику FC.

Конфигурация типичной сети FCIP представлена на рис. 2, из которого следует, что протокол FCIP практически не накладывает никаких ограничений ни на топологию, ни на протяженность сети.



В настоящее время многие организации реализуют сети FCIP, особенно для передачи данных в асинхронном режиме. Этому способствует, в частности, то, что FCIP полностью поддерживает весь набор оборудования и ПО, имеющегося для FC.

iFCP

iFCP (Internet Fibre Channel Protocol) – протокол, позволяющий объединять устройства с интерфейсами FC через IP-сети. Важное отличие от FCIP в том, что возможно объединять именно FC-устройства через IP-сеть, что позволяет для разной пары соединений иметь разный уровень QoS, что невозможно при туннелировании через FCIP.

Это протокол-шлюз, предназначенный для подключения построенных на Fibre Channel сетей хранения данных (SAN) и устройств SCSI. FCP представляет собой протокол Fibre Channel для SCSI. Он картирует команды SCSI и связанные с ними данные на транспортный уровень Fibre Channel FC-2. iFCP картирует существующий стандарт FCP и связанные с ним службы Fibre Channel на TCP/IP. Можно использовать даже при ненадежной работе основной сети IP.

Основное преимущество iFCP как протокола — вход в SAN заключается в картировании FCP на TCP, делая возможным сетевые (а не «точка-точка») соединения между сетями хранения. Таким образом, iFCP предлагает альтернативу для туннелирования фреймов Fibre Channel в TCP/IP. Для существующих драйверов и контроллеров FCP протокол iFCP предоставляет надежную транспортную среду между доменами SAN, и при этом никаких модификаций существующих продуктов не требуется.

Так как xFCP используют уровень FC layer 4, то можно ожидать интероперабельности с огромным числом существующих сегодня на рынке устройств и приложений SAN. Эта стратегия предоставляет возможность миграции от сегодняшней продукции к завтрашним сетям хранения данных, основанных на IP.

8. Технологии канального уровня и модель сетевой организации. Понятия инкапсуляции, конвергенции и туннелирования.

Для установления связи между двумя узлами сначала передается небольшой набор сигналов, используемых для синхронизации потока данных. После того, как соединение установлено, физические уровни обоих узлов сказываются связанными через среду передачи данных (например, через кабель), а их канальные уровни связаны логически благодаря используемым протоколам. Как только логический канал установлен, принимающий канальный уровень может декодировать сигнал и преобразовывать его в отдельные фреймы.

На канальном уровне выполняется проверка входящих сигналов, а также обнаруживаются повторно, неправильно или частично переданные данные во входящем потоке. При обнаружении ошибок уровень запрашивает у передающего узла повторную передачу данных – фрейм за фреймом. Для обнаружения ошибок на канальном уровне используется *контроль циклическим избыточным кодом* (cyclic redundancy check, CRC).

Канальный уровень содержит два важных подуровня: более высокий - *управление логическим соединением* (logical link control, LLC) и более низкий - *протокол управления доступом к передающей среде* (media access control, MAC). Подуровень LLC обеспечивает надежность коммуникаций путем установки канала передачи данных между двумя узлами и поддержки устойчивости этого канала. Подуровень MAC распознает *физический адрес* (или *адрес устройства)* иногда называемый *MAC-адресом,* содержащийся в каждом фрейме. Например, на некоторой рабочей станции подуровень MAC проверяет каждый фрейм, получаемый этой станцией, и передает фрейм более высокому уровню лишь в том случае, если адрес совпадает. В противном случае фрейм отбрасывается. Кроме того, подуровень MAC управляет совместной работой множества устройств внутри одной сети. В практическом задании 2-3 рассказывается о том, как определить адрес рабочей станции.

Сетевой уровень (network layer). Этот уровень управляет прохождением пакетов по сети. Все сети содержат физические маршруты передачи информации (кабельные тракты) и логические маршруты (программные тракты). Сетевой уровень анализирует адресную информацию протокола передачи пакетов и посылает их по наиболее подходящему маршруту – физическому или логическому, обеспечивая максимальную эффективность сети. Также этот уровень обеспечивает пересылку пакетов между сетями через маршрутизаторы.

Контролируя прохождение пакетов, сетевой уровень выступает в роли "управляющего трафиком": он маршрутизирует (направляет) пакеты по наиболее эффективному из нескольких возможных трактов передачи данных. Для определения наилучшего маршрута Сетевой уровень постоянно собирает информацию (метрики) о расположении различных сетей и узлов, этот процесс называется *обнаружением маршрута* (discovery).

Инкапсуляция — способ передачи данных между сетями разного типа или передачи нескольких протоколов с помощью одного протокола через различные сети. При инкапсуляции фрейм или пакет данных сети одного типа помещается в заголовок фрейма или пакета, применяемого в сети другого типа. При таком подходе новый заголовок выполняет функции почтового конверта для посланного письма, обеспечивая его соответствующей адресной и управляющей информацией, необходимой для того, чтобы послание достигло пункта назначения. В зависимости от их назначения, инкапсуляцию фреймов или пакетов выполняют сами компьютеры или сетевые устройства. Например, инкапсуляция используется для того, чтобы передать фрейм или пакет от одной локальной сети Microsoft или Novell – через Интернет – другой локальной сети аналогичного типа.

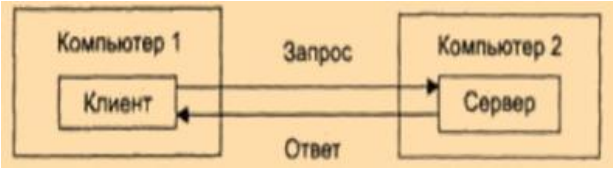
Конвергенция информационных технологий - процесс сближения разнородных электронных технологий в результате их быстрого развития и взаимодействия.

9. Клиент серверное взаимодействие. Виды соединий. Понятие широковещательной сети.

На тех компьютерах, ресурсы которых должны быть доступными всем пользователем сети, необходимо добавить модули, которые будут постоянно находится в режиме ожидания запросов, поступающих от других компьютеров.

На компьютерах, пользователи которых хотят получить доступ к ресурсам других компьютеров, также надо добавить к операционной системе некоторые специальные программные модули, которые должны вырабатывать запросы на доступ к удалённым ресурсам и передавать их по сети на нужный компьютер. Сетевые адаптеры и каналы связи решают в сети достаточно простую задачу- они передают сообщения с запросами и ответами от одного компьютера к другому, а основную работу по организации совместного использования ресурсов выполняют клиентские и серверные части операционной системы.

Пара модулей клиент- сервер обеспечивает совместный доступ пользователей к определённому типу ресурсов, например к файлам.



Сетевые службы всегда представляют собой распределенные программы. Распределенная программа - это программа, которая состоит из нескольких взаимодействующих частей (в приведенном на рис. 1.7 примере - из двух), причем каждая часть, как правило, выполняется на отдельном компьютере сети.

Сокеты — это объекты, которые инкапсулируют в себе все необходимые средства для обмена данными в сетях Интернет/интранет. Эти объекты сами создают соединения и посылают/принимают. Клиентский сокет предназначен для установления связи с сервером. Точнее, с серверным сокетом. Серверный сокет предполагает обмен данными с клиентами. От клиентской реализации он отличается тем, что способен работать с несколькими клиентами одновременно.

Широковещательная сеть- сеть, к которой подключено более двух маршрутизаторов, имеет возможность передать одно физическое сообщение всем подсоединенным к ней маршрутизаторам (т.е. осуществляется широковещание). Примером широковещательной сети является Ethernet .

10. Проектирование сетей: домены коллизий.

В технологии Ethernet, независимо от применяемого стандарта физического уровня, существует понятие домена коллизий.

Коллизия- 1 или более компов считают, что сеть свободна и начинают передавать инфу.

Домен коллизий (collision domain) - это часть сети Ethernet, все узлы которой конкурируют за общую разделяемую среду передачи и следовательно каждый узел которой может создать коллизию с любым другим узлом этой части сети. Сеть Ethernet, построенная на повторителях, всегда образует один домен коллизий. Мосты, коммутаторы и марщрутизаторы делят сеть ETHERNET на несколько доменов коллизий.

Если, например, столкновение кадров произошло в концентраторе 4, то в соответствии с логикой работы концентраторов 10Base-T сигнал коллизии распространится по всем портам всех концентраторов.

Если же вместо концентратора 3 поставить в сеть мост, то его порт С, связанный с концен-

тратором 4, воспримет сигнал коллизии, но не передаст его на свои остальные порты, так как это не входит в его обязанности. Мост просто отработает ситуацию коллизии средствами порта С, который подключен к общей среде, где эта коллизия возникла. Если коллизия возникла из-за того, что мост пытался передать через порт С кадр в концентратор 4, то, зафиксировав сигнал коллизии, порт С приостановит передачу кадра и попытается передать его повторно через случайный интервал времени. Если порт С принимал в момент возникновения коллизии кадр, то он просто отбросит полученное начало кадра и будет ожидать, когда узел, передававший кадр через концентратор 4, сделает повторную попытку передачи. После успешного принятия данного кадра в свой буфер мост передаст его на другой порт в соответствии с таблицей продвижения, например на порт А. Все события, связанные с обработкой коллизий портом С, для остальных сегментов сети, которые подключены к другим портам моста, просто останутся неизвестными.

Иерархическое соединение концентраторов Ethernet

Узлы, образующие один домен коллизий, работают синхронно, как единая распределенная электронная схема.

11. Проектирование сетей: Понятие СКС, основные конструктивы, методы монтажа, ограничения.

http://www.ysn.ru/docs/ckc/ckc/

СКС – структурированная кабельная система.

Структурированная кабельная система - это совокупность пассивного коммуникационного оборудования:

Кабель- этот компонент используется как среда передачи данных СКС. Кабель различают на экранированный и неэкранированный.

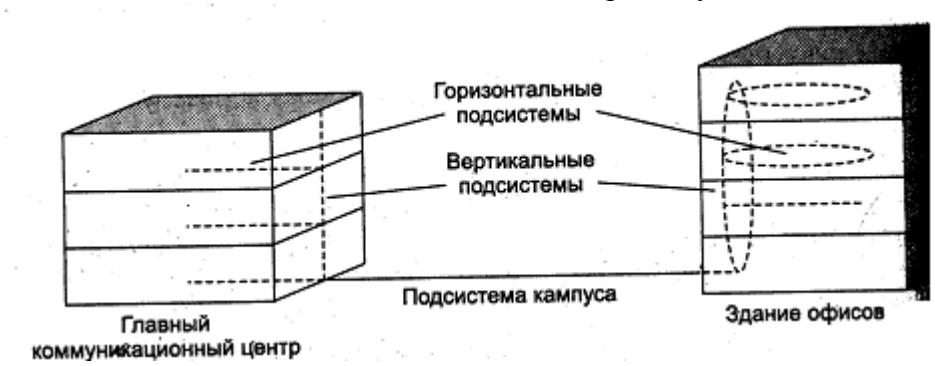
Розетки- этот компонент используют как точки входа в кабельную сеть здания.

Коммутационные панели- используются для администрирования кабельных систем в коммутационных центрах этажей и здания в целом.

Коммутационные шнуры- используются для подключения офисного оборудования в кабельную сеть здания, организации структуры кабельной системы в центрах коммутации.

СКС - охватывает все пространство здания, соединяет все точки средств передачи информации, такие как компьютеры, телефоны, датчики пожарной и охранной сигнализации, системы видеонаблюдения и контроля доступа. Все эти средства обеспечиваются индивидуальной точкой входа в общую систему здания. Линии, отдельные для каждой информационной розетки, связывают точки входа с коммутационным центром этажа, образуя **горизонтальную кабельную подсистему**. Все этажные коммутационные узлы специальными магистралями объединяются в коммутационном центре здания. Сюда же подводятся внешние кабельные магистрали для подключения здания к глобальным информационным ресурсам,

таким как телефония, интернет и т.п. Такая топология позволяет надежно управлять всей системой здания, обеспечивает гибкость и простоту системы.



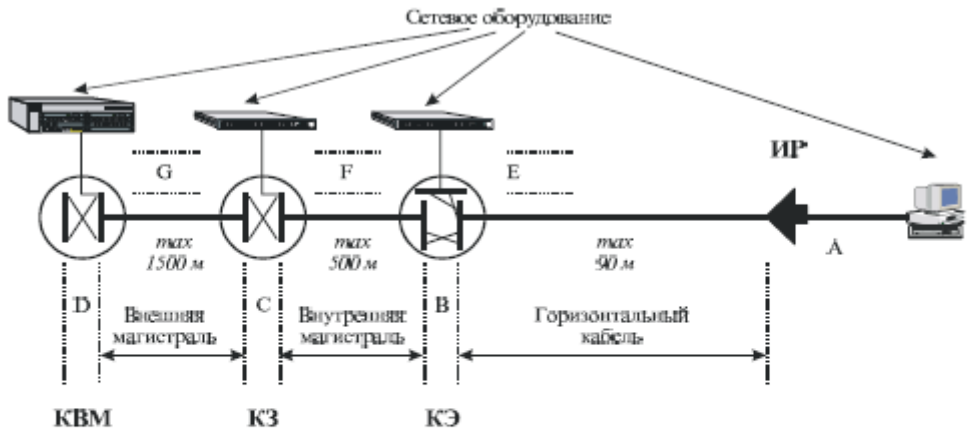
Составные части СКС на плане этажа:

- Горизонтальные подсистемы- этажи- соединят кроссовые этажи с розетками пользователей
- Вертикальные подсистемы соединяют кроссовые шкафы каждого этажа с центральной аппаратной здания
- Подсистемы кампуса объединяют несколько зданий с центральной аппаратной здания

Одним из способов повышения технико- экономической эффективности кабельных систем офисных зданий является минимизация типов кабелей, применяемых для их построения. В СКС используются только:

- Симметричные электрические кабели на основе витой пары с волновым сопротивлением 100, 120 и 150 Ом в экранированном и неэкранированном исполнении
- Одномодовых и многомодовых оптических кабелей

Ограничения на длины кабелей :



A+B+E ≤ 10 м — суммарная длина всех шнуров и перемычек горизонтальной подсистемы;
C и D ≤ 20 м — длина коммутационных шнуров (перемычек) в КЗ и КВМ;
F и G ≤ 30 м — длина оконечных шнуров в КЗ и КВМ.

Примечания:
1. Все указанные длины — физические длины.
2. Длины 10 м (A+B+E) и 30 м (F и G) являются рекомендуемыми.

Рис. 4. Максимальные расстояния в кабельной системе по ISO/IEC 11801

Таблица 11. Максимальные длины кабельных трасс в зависимости от типа кабеля и класса приложения

Класс приложений	A	B	C	D	Оптики
Среда передачи сигнала					
Симметричный кабель категории 3	2 км	200 м	100 м [1)]		
Симметричный кабель категории 4	3 км	260 м	150 м		
Симметричный кабель категории 5	3 км	260 м	160 м	100 м	
Симметричный кабель 150 Ом	3 км	400 м	250 м	150 м	
Многомодовый оптический кабель	-	-	-	-	2 км
Одномодовый оптический кабель	-	-	-	-	3 км [2)]

Примечания:

1. Под длиной 100 м понимается суммарная длина горизонтального кабеля (до 90 м) и соединительных шнуров.
2. 3 км — ограничение, формально наложенное стандартом. Не является физическим ограничением для одномодовых волоконных световодов.

В перечень основных видов работ, выполняемых в процессе монтажа СКС, входит входной контроль отдельных компонентов, прокладка кабелей магистральных и горизонтальных подсистем, монтаж декоративных коробов и 19-дюймового конструктива, подключение кабелей к розеткам и информационным панелям. Порядок выполнения тех или иных видов работ в значительной степени определяется условиями строительной готовности здания.

Заключительными этапами монтажа СКС являются тестирование, подключение сетевой аппаратуры, коммутация каналов передачи информации и заполнение кабельного журнала.

Правила монтажа:

Неэкранированный кабель. Для избежания растяжения кабеля во время монтажа сила натяжения не должна превышать 110 Н для 4-парных кабелей калибра 24 AWG (0,5 мм). При монтаже кабельных систем в сложных условиях (внутри или между зданиями), при протяженности непрерывного кондуита более 30 м или сумме углов поворотов при протяжке, превышающей 180 градусов, рекомендуется применять динамометр, позволяющий контролировать натяжение кабеля с целью не выйти за рамки спецификаций производителя.

Запрещается помещать кабели в те каналы, кабинеты, корпуса и другие монтажные устройства, у которых радиусы закруглений или краев не соответствуют требованиям производителей кабелей к радиусу их изгиба.

К мерам предосторожности, соблюдаемым при монтаже и организации кабельных потоков, относится предотвращение различных механических напряжений в кабеле, вызываемых натяжением, резкими изгибами и чрезмерным стягиванием пучков кабеля.

Следует избегать негативных воздействий на кабель, вызываемых: его перекручиванием во время протягивания или монтажа, растягиванием кабельных пучков под действием собственного веса на кабельных подвесках, туго затянутыми кабельными хомутами, резкими изгибами кабеля.

Экранированный кабель. Экран кабельной линии или канала должен быть заземлен на шине телекоммуникационной системы заземления (Telecommunications Ground Busbar, TGB). Разница потенциалов между экраном и землей не должна превышать 1 В, а сопротивление между экраном и землей - 4 Ом на рабочем месте. Для создания магистрали между двумя зданиями с различными потенциалами земли рекомендуется использовать волоконно-оптические кабели.

Экраны всех кабелей должны быть заземлены в телекоммуникационном шкафу. Путь к "земле" должен быть постоянным и непрерывным. Экран кабелей должен обеспечивать непрерывный путь к "земле" во всех частях экранированной кабельной системы. Для снижения сопротивления заземления рекомендуется соединять металлические кондуиты с проводниками системы заземления, проходящими в них, на обоих концах кондуита.

12. Проектирование сетей: трассировка кабельных трасс.

Масса времени тратится на поиск места залегания (трассировку) проложенного кабеля, проводов (линий), а также места нахождения шкафа и конкретного элемента коммутационного оборудования, к которому они подключены (идентификацию окончаний). Даже если кабель виден непосредственно, то проследить его путь среди толстого пучка других кабелей - непростая задача. Не меньше времени может отнять и поиск нужной пары проводников в кабеле, проверка целостности цепей, поиск выключателя розеток питающей сети и т. д. Выполнение перечисленных операций не займет много времени, если вы имеете недорогие приборы, обзор которых приведен ниже. Они применимы на любых типах кабельных линий, а приобрести необходимые для быстрого выполнения операций навыки не составит труда.

Для трассировки и идентификации окончаний кабелей, проводников и кабельных каналов достаточно иметь тональный генератор и индуктивный щуп. Принцип действия этих приборов - поиск трассируемого кабеля или канала по наведенному в нем сигналу. Сигнал, формируемый специальным генератором, подается на кабель в любом доступном месте. Щуп обеспечивает прием сигнала датчиком, его усиление и воспроизведение через динамик или наушники. Таким образом, по уровню громкости сигнала монтажник может определить место залегания кабеля и проследить трассу вдоль линии, начиная с места подачи сигнала. Естественно, генератор и щуп должны иметь аналогичные параметры. Кроме того, их характеристики должны соответствовать и типу трассируемых линий: кабелей внутри зданий, подземных кабельных линий, силовых линий, металлических каналов.

Самыми важными параметрами генераторов являются мощность, характер наводимого сигнала (постоянная частота, две чередующиеся частоты, импульсы постоянного напряжения) и значение частоты. Реализуемые генератором способы подачи сигнала в трассируемую линию не менее важны. Так сигнал может подаваться:

- непосредственно на жилы одного из концов кабеля (например, в распределительной коробке) с помощью зажимов типа "крокодил";
- на весь кабель без нарушения его оболочки (в местах, где он доступен в коробах, колодцах, шахтах) с помощью индуктивного хомута, охватывающего кабель;

- на кабель под землей от антенны, расположенной над ним на поверхности (для трассировки длинных подземных кабелей).

Кроме того, специальные модели приборов позволяют произвести трассировку коаксиальных и воздушных кабельных линий.

ТРАССИРОВКА КАБЕЛЬНЫХ ЛИНИЙ И КАНАЛОВ

Эффективность работы индуктивного щупа может быть выше, если он имеет комбинированный датчик. Штыревая антенна обеспечивает более высокую чувствительность в случаях, когда сигнал генератора подается на отключенные или замкнутые на высокоомную нагрузку жилыПоиск кабеля в длинном пучке может оказаться затруднен из-за наводок сигнала в других кабелях пучка. В таких случаях очень удобны щупы с регулятором чувствительности, при соответствующей настройке которых слабый сигнал не будет восприниматься. Не менее удобен в подобных ситуациях и линейный визуальный индикатор, дающий более точное представление об уровне сигнала.

ТРАССИРОВКА И ИДЕНТИФИКАЦИЯ ЦЕПЕЙ ПИТАНИЯ

Особый случай - трассировка цепей питания. Отключить их не всегда возможно - в частности из-за риска обесточить вместе с нужной цепью еще несколько, подключение которых к выключателю не было отражено в схемах.
Определить место залегания кабеля сети переменного тока 220 В 50 Гц можно с помощью любого индуктивного щупа без фильтра-пробки. Однако такой способ годится лишь для предотвращения повреждений скрытых кабелей при изготовлении отверстий в стенах, так как не позволяет отличить одну цепь от другой.
Поэтому трассировка и идентификация цепей питания (от 9 до 600 В) без их отключения производится с применением дополнительного генератора. Он включается в розетку или параллельно основной нагрузке и представляет собой сопротивление, изменяющее свое значение с увеличением частоты. Подаваемый сигнал не влияет на работу подключенных к трассируемой линии устройств. В то же время изменение тока в подключенной к генератору линии позволяет без труда
произвести трассировку цепи питания, начиная с розеток или нагрузки, и идентифицировать выключатели на силовых щитах, к которым они подключены. Кроме того, трассировку линий под напряжением можно выполнить с помощью индуктивного хомута. Разница между этими двумя способами заключается в том, что первый позволяет трассировать линии от места подключения генератора в сторону источника напряжения, а второй - от места подключения в сторону нагрузки.

13. Проектирование сетей: Концепция сетевой безопасности: аутентификация, целостность сообщений, конфиденциальность с помощью симметричного шифрования, ассиметричный общедоступный ключ шифрования, комбинированное шифрование.

http://www.bytemag.ru/?ID=601945
http://www.bytemag.ru/?ID=602031

Безопасная информационная система- это система, которая, во-первых, защищает данные от несанкционированного доступа, во-вторых, всегда готова предоставить их всем пользователям, а в-третьих надёжно хранит информацию и гарантирует неизменность данных.
Аутентификация предотвращает доступ к сети нежелательных лиц и разрешает вход для легальных пользователей. В переводе этот термин означает “установление подлинности”.
Целостность- гарантия сохранности данными правильных значений, которая обеспечивается запретом для неавторизованных пользователей каким- либо способом изменить, модифицировать, разрушать или создавать данные.

Шифрование

Процесс преобразования открытых данных в зашифрованные и наоборот принято называть шифрованием, причем две составляющие этого процесса называют соответственно зашифрованием и расшифрованием.
Ключ представляет собой уникальный элемент, с помощью которого можно изменять результаты работы алгоритма шифрования: один и тот же исходный текст при использовании различных ключей будет зашифрован по-разному.
Алгоритмы шифрования можно разделить на две категории: симметричного и асимметричного шифрования.

Симметричное шифрование

Симметричное шифрование: посторонним лицам известен алгоритм шифрования, но он зависит от небольшой порции секретной информации - ключа, одинакового для отправителя и получателя сообщения;

Алгоритм, определяемый ГОСТ 28147-89 (рис. 1), имеет длину ключа шифрования 256 бит. Он шифрует информацию блоками по 64 бит (такие алгоритмы называются блочными), которые затем разбиваются на два субблока по 32 бит (N1 и N2). Субблок N1 обрабатывается определенным образом, после чего его значение складывается со значением субблока N2 (сложение выполняется по модулю 2, т. е. применяется логическая операция XOR - "исключающее или"), а затем субблоки меняются местами. Данное преобразование выполняется определенное число раз ("раундов"): 16 или 32 в зависимости от режима работы алгоритма. В каждом раунде выполняются две операции.

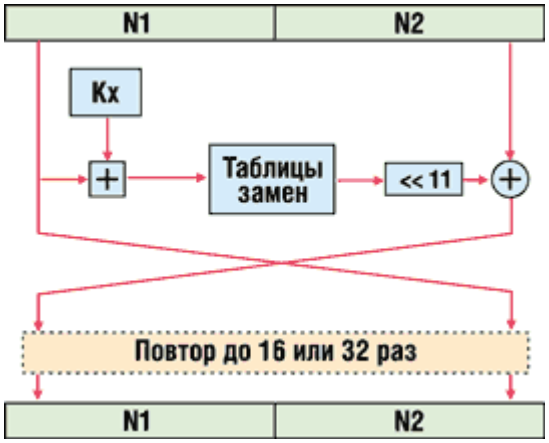


Рис. 1. Схема алгоритма ГОСТ 28147-89.

Первая - наложение ключа. Содержимое субблока N1 складывается по модулю 2 с 32-бит частью ключа Kx. Полный ключ шифрования представляется в виде конкатенации 32-бит подключей: K0, K1, K2, K3, K4, K5, K6, K7. В процессе шифрования используется один из этих подключей - в зависимости от номера раунда и режима работы алгоритма.

Вторая операция - табличная замена. После наложения ключа субблок N1 разбивается на 8 частей по 4 бит, значение каждой из которых заменяется в соответствии с таблицей замены для данной части субблока. Затем выполняется побитовый циклический сдвиг субблока влево на 11 бит.

Табличные замены (Substitution box - S-box) часто используются в современных алгоритмах шифрования, поэтому стоит пояснить, как организуется подобная операция. В таблицу записываются выходные значения блоков. Блок данных определенной размерности (в нашем случае - 4-бит) имеет свое числовое представление, которое определяет номер выходного значения. Например, если S-box имеет вид 4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1 и на вход пришел 4-бит блок "0100" (значение 4), то, согласно таблице, выходное значение будет равно 15, т. е. "1111" (0 а 4, 1 а 11, 2 а 2 ...).

Асимметричное шифрование

Алгоритмы асимметричного шифрования, как уже отмечалось, используют два ключа: k1 - ключ зашифрования, или открытый, и k2 - ключ расшифрования, или секретный. Открытый ключ вычисляется из секретного: k1 = f(k2).

Комплексный метод

Этот метод применения алгоритмов симметричного и асимметричного шифрования устраняет ряд недостатков, свойственных каждому из них при раздельном применении. Итак, чтобы обменяться зашифрованными сообщениями через Интернет, пользователям I и J необходимо предварительно сделать следующее.

1. Выбрать алгоритмы шифрования и их параметры. Каждый алгоритм имеет множество параметров, которые должны быть идентичны, - иначе даже при наличии правильного ключа шифрования будет невозможно расшифровать информацию.
2. Сгенерировать свои ключи асимметричного шифрования. Пользователь I генерирует пару ключей KsI (secret - секретный) и KpI (publIc - открытый), пользователь J создает пару KsJ и KpJ.
3. Обменяться открытыми ключами или сделать их доступными друг другу. Предположим, что открытые ключи KpI и KpJ пользователи пересылают друг другу по электронной почте.
Предположим, ключи успешно переданы и получены, после чего можно отправлять зашифрованное сообщение. Это и делает пользователь J, отправляя пользователю I сообщение M.
Процесс обмена иллюстрирует рис. 1. Перед передачей сообщения пользователь J создает случайный ключ симметричного шифрования (назовем его Ksimm). Пользователь J асимметрично зашифровывает ключ Ksimm на открытом ключе пользователя I KpI и отправляет зашифрованный ключ пользователю I. Затем пользователь J зашифровывает ключом Ksimm сообщение M, и полученный шифртекст C отсылается пользователю I. Пользователь I получает зашифрованный ключ Ksimm и асимметрично расшифровывает его своим секретным ключом KsI. С помощью полученного ключа Ksimm пользователь I расшифровывает сообщение M.

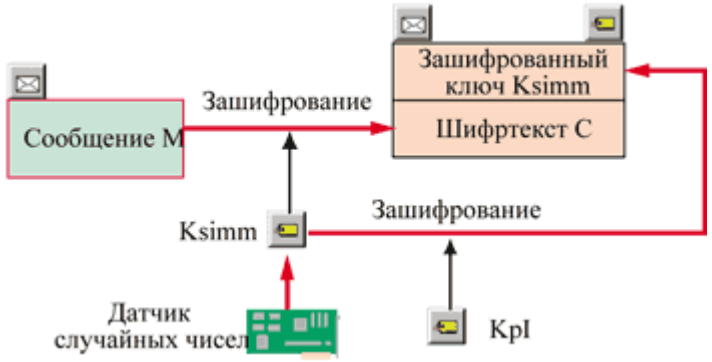


Рис. 1. Комплексный метод шифрования.

Скрытое распространение ключей симметричного шифрования достигается за счет того, что симметричный ключ Ksimm, на котором шифруется собственно информация, передается по открытым каналам связи в зашифрованном виде - для его зашифрования используется асимметричный алгоритм, не имеющий проблем с секретностью. Проблемы малой скорости асимметричного шифрования в данном случае практически не возникает, поскольку асимметричным алгоритмом шифруется только короткий ключ Ksimm, а все данные шифруются быстрым симметричным алгоритмом. Результат - быстрое шифрование в сочетании с удобным обменом ключами.

14. Протокол PPP: характеристики, сжатие в PPP, аутентификация, автоматичсекое отслеживание качества связи.

Протокол "точка-точка" (PPP) — набор стандартных протоколов, обеспечивающих взаимодействие программного обеспечения удаленного доступа от различных поставщиков.
Связь по протоколу PPP состоит из четырёх стадий:

- установление связи (осуществляется выбор протоколов аутентификации, шифрования, сжатия и устанавливаются параметры соединения),
- установление подлинности пользователя (реализуются алгоритмы аутентификации, на основе протоколов (PAP, SPAP, CHAP, MS-CHAP, MS-CHAP v2 и EAP),
- контроль повторного вызова PPP (необязательная стадия, в которой подтверждается подлинность удалённого клиента),

- вызов протокола сетевого уровня (реализация протоколов установленных в первой стадии).

. Большинство реализаций PPP позволяет полностью автоматизировать последовательность входа в систему.

Методы аутентификации

Протокол PPP PAP

PAP не является сильным аутентификационным методом. PAP аутентифицирует только вызывающего оператора, а пароли пересылаются по каналу, который считается уже "защищенным". Таким образом, этот метод не дает защиты от использования чужих паролей и неоднократных попыток подбора пароля. Частота и количество неудачных попыток входа в сеть контролируются на уровне вызывающего оператора.

Протокол PPP CHAP

CHAP используется для периодической аутентификации центрального компьютера или конечного пользователя с помощью согласования по трем параметрам. Аутентификация происходит в момент установления связи, но может повторяться и после ее установления.
На рисунке 1 показан процесс аутентификации CHAP. Маршрутизатор и сервер доступа используют общий секретный ключ "trustme".
Маршрутизатор отделения пытается провести аутентификацию сервера сетевого доступа (NAS) или "аутентификатора". CHAP обеспечивает безопасность сети, требуя от операторов обмена "текстовым секретом". Этот секрет никогда не передается по каналу связи. По завершении этапа установления связи аутентификатор передает вызывающей машине запрос, который состоит из аутентификатора (ID), случайного числа и имени центрального компьютера (для местного устройства) или имени пользователя (для удаленного устройства). Вызывающая машина проводит вычисления с помощью односторонней хэш-функции. Аутентификатор, случайное число и общий "текстовый секрет" один за другим подаются на вход хэш-функции. После этого вызывающая машина отправляет серверу ответ, который состоит из хэша и имени центрального компьютера или имени пользователя удаленного устройства. По получении ответа аутентификатор проверяет проставленное в ответе имя и выполняет те же вычисления. Затем результат этих вычислений сравнивается с величиной, проставленной в ответе.
Если эти величины совпадают, результат аутентификации считается положительным, система выдает соответствующее уведомление, и LCP устанавливает связь. Секретные пароли на местном и удаленном устройстве должны быть идентичны. Поскольку "текстовый секрет" никогда не передается по каналам связи, никто не может подслушать его с помощью каких-либо устройств и использовать для нелегального входа в систему. Пока сервер не получит адекватный ответ, удаленное устройство не сможет подключиться к местному устройству.
CHAP обеспечивает защиту от использования чужих паролей за счет пошаговых изменений аутентификатора и применения переменной величины запроса. Повторяющиеся запросы предназначены для ограничения времени, в течение которого система теоретически остается подверженной любой отдельной хакерской атаке. Частоту и количество неудачных попыток входа в систему контролирует аутентификатор.

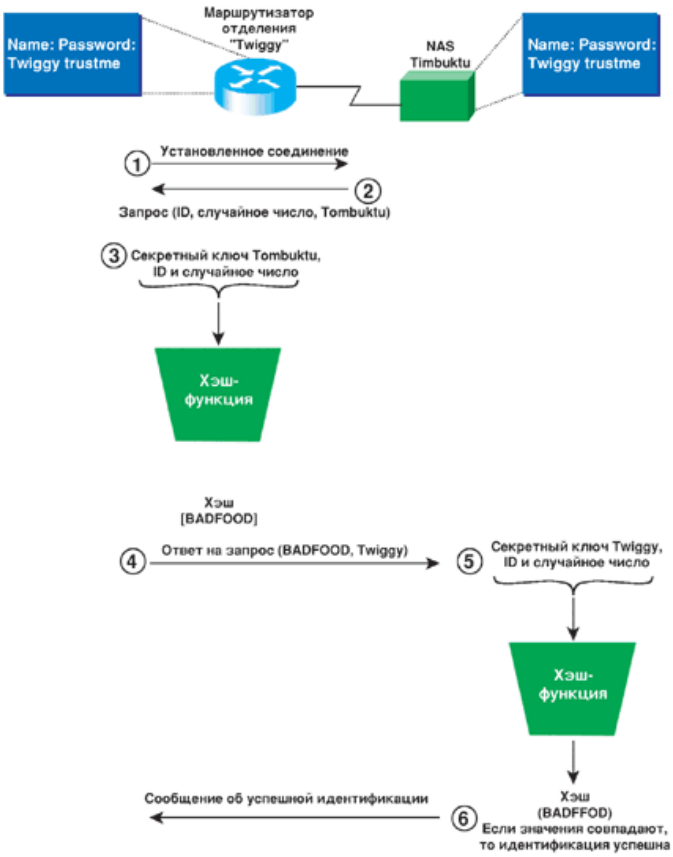


Рисунок 1. Аутентификация PPP CHAP

Протокол PPP EAP

PPP EAP является общим протоколом аутентификации PPP, который поддерживает множество аутентификационных механизмов. EAP не производит выбор конкретного аутентификационного механизма на этапе контроля соединения, но откладывает этот выбор до этапа аутентификации. Этот сценарий позволяет аутентификатору запросить больше информации до определения конкретного аутентификационного механизма. Кроме того, это дает возможность использовать "внутренний" сервер, который реально запускает различные механизмы, тогда как аутентификатор PPP служит лишь для обмена аутентификационными данными.

Рисунок 2. Аутентификация PPP EAP

Маршрутизатор отделения пытается провести аутентификацию сервера сетевого доступа (NAS) или “аутентификатора”. По завершении этапа установления связи аутентификатор отправляет один или несколько запросов для аутентификации вызывающей машины. В запросе имеется поле типа запроса, где указано, что именно запрашивается. Так, например, здесь можно указать такие типы запросов, как аутентификация MD5, S/Key, аутентификация с использованием аппаратной карты для генерирования паролей и т.д. Запрос типа MD5 очень схож с протоколом аутентификации CHAP. Обычно аутентификатор отправляет первоначальный аутентификационный запрос, за которым следует один или несколько дополнительных запросов о предоставлении аутентификационной информации. При этом первоначальный запрос не является обязательным и может опускаться в случаях, когда аутентификация обеспечивается иными способами (при связи по выделенным каналам, выделенным номерам и т.д.). В этих случаях вызывающая машина отправляет пакет ответных данных в ответ на каждый запрос.

Сжатие данных

Еще одной чертой PPP является контроль и восстановление ошибок. Для повышения эффективности работы по медленным линиям протокол управления каналом допускает удаление постоянного флажка и полей адреса, а также сокращение поля протокола с двух байт до одного. Таким образом, PPP-пакеты по крайней мере на три байта длиннее, чем SLIP-пакет. Сжатие заголовка PPP-пакета позволяет повышать скорость связи и хотя бы частично компенсировать больший размер пакета. Какие данные сжимаются и сжимаются ли вообще, определяется специальными правилами.

Отслеживание качества связи. ???

Фаза переговоров во время PPP-соединения делает протокол PPP уникальным среди других сетевых протоколов. Так, его предшественник, протокол последовательного соединения с Internet (Serial Line Internet Protocol — SLIP) поддерживал передачу только протокола IP через телефонную линию. Если протокол PPP поддерживается устройствами на обоих концах линии, то он позволяет виртуально передавать через модемное соединение любой сетевой трафик. После проверки подлинности соединения клиентское устройство PPP должно определить, какие протоколы оно будет передавать по этому соединению. Далее удаленный хост PPP обязан принять или отвергнуть протоколы, которые им не поддерживаются. Для переговоров о возможности передачи тех или иных протоколов через PPP-соединение используется протокол управления сетью (Network Control Protocol — NCP). В течение переговоров хост и клиент должны согласовать параметры для передачи каждого протокола. Например, в течение переговоров о передаче протокола IP-хост снабжает клиента информацией об IP-адресе и сервере DNS, чтобы клиент мог нормально обращаться к сети хоста PPP. После того как переговоры о протоколе успешно завершены, между хостом и клиентом начинается передача пакетов данных.

Когда клиент желает завершить сеанс PPP, то организуется еще один сеанс по протоколу LCP с целью завершения соединения. Сервер PPP должен правильно опознать этот запрос и закрыть модемное соединение с клиентом PPP.

15. Конфигурация сетей с помощью BOOTP и DHCP.

BOOTP

BOOTP (Bootstrap Protocol) — это сетевой протокол, который обеспечивает определение с помощью специального севера IP- адреса клиента по его MAC адресу, а также позволяет клиентам узнавать другие параметры загрузки (например, имя программы, загружаемой затем с помощью TFTP) и использует UDP вместо протокола канального уровня. Это позволяет использовать маршрутизаторы (bootp relay) для передачи запросов и ответов из одного сегмента локальной сети в другой. Протокол DHCP (Dynamic Host Configuration Protocol) является надстройкой над BOOTP (для совместимости с bootp relay) и позволяет серверу выделять IP-адреса клиентам динамически на ограниченный срок. Порт сервера — UDP/67 (BOOTPS), клиента — UDP/68 (BOOTPC). Клиент делает широковещательный

(255.255.255.255 — всем в локальной сети, номера которой я не знаю) запрос bootrequest (один нефрагментированный пакет): обязательно содержит аппаратный MAC-адрес клиента и может содержать преполагаемый IP-адрес клиента, имя сервера и обобщённое имя файла для загрузки. Сервер отвечает пакетом bootreply (обычно unicast, так как MAC- и IP-адреса клиента ему известны): IP-адрес клиента, обобщённое имя файла замещается на полное имя файла исходя из конфигурации сервера, типа и адреса клиента и др. Собственно загрузка файла осуществляется клиентом с помощью протокола TFTP. Клиент должен быть в состоянии ответить на ARP запросы, чтобы мог работать TFTP-сервер
Поскольку протоколы BOOTP и DHCP взаимосвязаны, они имеют общие определяющие характеристики. Общие элементы перечислены ниже.

DHCP

Требовался механизм, который позволил бы ликвидировать стадию ручного конфигурирования компьютеров, поддерживал многосегментные сети, не требуя наличия DHCP-сервера в каждой подсети, не конфликтовал с существующими сетевыми протоколами и компьютерами, имеющими статичную конфигурацию, был способен взаимодействовать с ретранслирующими агентами протокола BOOTP и обслуживать BOOTP-клиентов, наконец, допускал управление передаваемыми параметрами конфигурации. Что касается более узких задач, то DHCP должен был обеспечивать уникальность сетевых адресов, используемых разными компьютерами сети в данный момент, сохранение прежней конфигурации клиентской станции после перезагрузки клиента или сервера, автоматическое присвоение параметров конфигурации вновь подключенным машинам.

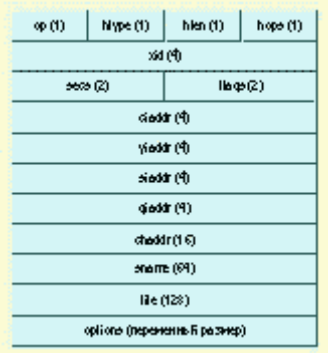


Рис. 1. Формат сообщения DHCP (в скобках - размер поля в байтах)

Поле	**Описание**
op	Тип сообщения (1 = BOOTREQUEST, 2 = BOOTREPLY)
htype	Тип адреса оборудования
hlen	Длина адреса оборудования
hops	Используется ретранслирующим агентом
xid	Идентификатор транзакции между сервером и клиентом
secs	Время с момента выдачи DHCPREQUEST или начала обновления конфигурации
flags	Флаги (первый бит маркирует широковещательные сообщения)
ciaddr	IP-адрес клиента
yiaddr	<Ваш> (клиентский) IP-адрес
siaddr	IP-адрес следующего сервера, участвующего в загрузке

giaddr	IP-адрес ретранслирующего агента
chaddr	<Аппаратный> адрес клиента
sname	Хост-имя сервера (опция)
file	Имя загрузочного файла
options	Поле дополнительных параметров

Принципы архитектуры и формат сообщений

Работа протокола DHCP базируется на классической схеме клиент-сервер. В роли клиентов выступают компьютеры сети, стремящиеся получить IP-адреса в так называемую аренду (lease), а DHCP-серверы выполняют функции диспетчеров, которые выдают адреса, контролируют их использование и сообщают клиентам требуемые параметры конфигурации. Сервер поддерживает пул свободных адресов и, кроме того, ведет собственную регистрационную базу данных. Взаимодействие DHCP-серверов со станциями-клиентами осуществляется путем обмена сообщениями.

Сравнивая протоколы BOOTP и DHCP, нельзя не отметить появления в DHCP новых услуг. Во-первых, в этом протоколе предусмотрен механизм автоматической выдачи IP-адресов во временное пользование с возможностью их последующего присвоения новым клиентам. Во-вторых, клиент может получить от сервера все параметры конфигурации, которые ему необходимы для успешного функционирования в IP-сети.

Указанные отличия потребовали частичного расширения формата сообщений. Так, в нем появилось отдельное поле идентификатора клиента, сделана более прозрачной интерпретация адреса сервера (поле siaddr), переменный размер получило поле options, используемое, в частности, для передачи параметров конфигурации (его длина обычно находится в диапазоне 312-576 байт, хотя возможно и дополнительное расширение этого поля за счет полей sname и file).

В роли транспортного протокола для обмена DHCP-сообщениями выступает UDP. При отправке сообщения с клиента на сервер используется 67-й порт DHCP-сервера, при передаче в обратном направлении - 68-й. Эти номера портов, как и схожая структура сообщений, обеспечивают обратную совместимость DHCP с BOOTP. Конкретные процедуры взаимодействия клиентов и серверов BOOTP и DHCP регламентирует документ RFC 1542.

Параметры конфигурации

Хранение параметров сетевой конфигурации станций-клиентов является второй услугой, предоставляемой DHCP-сервером. В создаваемой базе данных на каждого клиента заводится отдельная запись с уникальным ключом-идентификатором и строкой конфигурационных параметров.

Роль идентификатора может играть пара <номер подсети IP, аппаратный адрес>, которая позволит использовать аппаратный адрес сразу в нескольких подсетях, либо пара <номер подсети IP, имя хост-компьютера>, позволяющая серверу взаимодействовать с клиентом, перемещенным в другую подсеть.

Что касается собственно параметров конфигурации, то их набор, поддерживаемый протоколом DHCP, определен в спецификациях RFC 1122, 1123, 1196 и 1256. В него входят выдан-

ный адрес, срок его аренды, назначавшиеся ранее адреса, а также максимальный размер реассемблируемого пакета, перечень фильтров для нелокальной маршрутизации от источника, адрес, используемый в широковещательных пакетах, параметры статических маршрутов и т.д. Впрочем, из всей совокупности допустимых параметров (а их более 30) в процессе инициализации могут передаваться только те, которые действительно необходимы для работы клиента либо определяются спецификой конкретной подсети.

Редукция объема передаваемых сведений о конфигурации достигается двумя способами. Во-первых, для большей части параметров в упомянутых выше документах RFC определены значения, принимаемые по умолчанию. Клиент будет использовать их, если в сообщении, поступившем от сервера, какие-то параметры опущены. Во-вторых, отправляя сообщение DHCPDISCOVER или DHCPREQUEST, клиентская станция может явно указать в нем параметры, значения которых она хотела бы получить.

Очевидно, что в обоих случаях передача параметров конфигурации осуществляется в ходе основной процедуры выделения IP-адреса. Возможен, однако, случай, когда клиент уже имеет IP-адрес (например, он был задан вручную). Тогда он может выдать сообщение DHCPINFORM*, содержащее уже имеющийся адрес и запрос об отдельных параметрах конфигурации. Получив это сообщение, DHCP-сервер проверяет правильность адреса клиента (но не наличие аренды) и направляет ему сообщение DHCPACK с требуемыми параметрами конфигурации.

16. Протоколы Ethernet. Общие понятие, определения и термины, особенности.

В сетях Ethernet используется метод доступа к среде передачи данных, называемый методом коллективного доступа с опознаванием несущей и обнаружением коллизий (carrier-sense-multiply-access with collision detection, CSMA/ CD).

Этот метод применяется исключительно в сетях с логической общей шиной (к которым относятся и радиосети, породившие этот метод). Все компьютеры такой сети имеют непосредственный доступ к общей среде, поэтому она может быть использована для передачи данных между любыми двумя узлами сети. Одновременно все компьютеры сети имеют возможность немедленно (с учетом задержки распространения сигнала по физической среде) получить данные, которые любой из компьютеров начал передавать в общую среду.

Простота схемы подключения — один из факторов, определивших успех стандарта Ethernet. Говорят, что среда, к которой подключены все станции, работает в режиме коллективного доступа (Multiply Access, MA).

На MAC-уровне для идентификации сетевых интерфейсов узлов сети используются регламентированные стандартом IEEE 802.3 уникальные 6-байтовые адреса, называемые MAC-адресами. Каждый сетевой адаптер имеет, по крайней мере, один MAC-адрес.

Чтобы получить возможность передавать кадр, интерфейс-отправитель должен убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая также называется несущей частотой (carrier-sense, CS). Признаком незанятости среды является отсутствие на ней несущей частоты, которая при манчестерском способе кодирования равна 5-10 МГц, в зависимости от последовательности единиц и нулей, передаваемых в данный момент. Узел 1 обнаружил, что среда свободна, и начал передавать свой кадр. В классической сети Ethernet на коаксиальном кабеле сигналы передатчика узла 1 распространяются в обе стороны, так что все узлы сети их получают.

Все станции, подключенные к кабелю, могут распознать факт передачи кадра, и та станция, которая узнает собственный адрес в заголовках кадра, записывает его содержимое в свой внутренний буфер, обрабатывает полученные данные, передает их вверх по своему

стеку, а затем посылает по кабелю кадр-ответ. Адрес станции-источника содержится в исходном кадре, поэтому станция-получатель знает, кому нужно послать ответ.

Узел 2 во время передачи кадра узлом 1 также пытался начать передачу своего кадра, однако обнаружил, что среда занята - на ней присутствует несущая частота, - поэтому узел 2 вынужден ждать, пока узел 1 не прекратит передачу кадра.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу (Inter Packet Gap) в 9,6 мкс. Эта пауза, называемая также межкадровым интервалом, нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одной станцией. После окончания технологической паузы узлы имеют право начать передачу своего кадра, так как среда свободна. Из-за задержек распространения сигнала по кабелю не все узлы строго одновременно фиксируют факт окончания передачи кадра узлом 1.

При описанном подходе возможна ситуация, когда две станции одновременно пытаются передать кадр данных по общей среде. Механизм прослушивания среды и пауза между кадрами не гарантируют исключения такой ситуации, когда две или более станции одновременно решают, что среда свободна, и начинают передавать свои кадры. Говорят, что при этом происходит коллизия (collision), так как содержимое обоих кадров сталкивается на общем кабеле и происходит искажение информации - методы кодирования, используемые в Ethernet, не позволяют выделять сигналы каждой станции из общего сигнала.

Формат фрейма

Данные, передаваемые в стандарте Ethernet, помещаются во фреймы (рис. 2.11). Каждый фрейм состоит из строго определенных фрагментов (полей). Первый фрагмент – заголовок (preamble), имеет длину 56 бит. Заголовок синхронизирует передачу фрейма и состоит из перемещающейся последовательности нулей и единиц. Следующее поле – 8-битный разграничитель фреймов (называемый SFD или SOF). Признак начала фрейма имеет значение 10101011 и указывает на то, что далее во фрейме следует адресная информация. За этим признаком помещаются два адресных поля, содержащих адреса назначения и источника. Согласно рекомендациям IEEE 802.3, адресные поля могут иметь длину 16 или 48 бит (обычно 48). Имеются два адреса: адрес источника (source address, SA), представляющий собой адрес передающего узла, и адрес назначения (destination address, DA), являющийся адресом принимающего узла. Далее 16-битное поле указывает длину поля данных (идущего следом).

Заголовок	SFD	Адрес назначения	Адрес источника	Длина	Данные и поле-заполнитель	FCS
56	8	16 или 48	16 или 48	16	368—12000	32

Рис. 2.11. Побитовое представление формата фрейма 802.3

Раздел данных во фрейме идет вслед за полем длины. Длина инкапсулированных данных должна быть кратна 8 (одному байту). Если реальные данные имеют длину менее 368 бит или не кратны 8, добавляется поле-заполнитель. Длина поля данных с заполнителем может быть от 368 до 12 000 бит (или от 46 до 1500 байт). Последний фрагмент фрейма – поле контрольно последовательности (суммы) фрейма (frame check sequence, FCS), имеющее длину 32 бита. Для обнаружения ошибок это поле содержит значение дм контроля с помощью циклического избыточного кода (CRC). Это значение вычисляется на основе значений других полей фрейма в момент инкапсуляции данных. При приеме фрейма он пересчитывается заново. Если результат повторного вычисления не совпадает с исходным, генерируется ошибка и принимающий узел запрашивает повторную передачу данного фрейм. Если результаты вычислений совпадают, алгоритм получения контрольно суммы указывает на то, что повторная передача не требуется. Алгоритм CRC определяется стандартом IEEE.

Ethernet II – метод форматирования фреймов Ethernet, используемый в Интернете и других современных сетях, немного отличающихся от традиционного стандарта IEEE 802.3 (однако в настоящее время признанный часть стандарта IEEE 802.3 и описанный в RFC 894), для повышения эффективности сетевых коммуникаций. В фрейме Ethernet II заголовок имеет длим 64 бита и содержит как информацию для синхронизации фреймов, так и признак начала фрейма (SOF). Адреса назначения и источника во фрейм Ethernet II имеют длину точно 48 бит, как показано на рис. 2.12.

Заголовок и SOF	Адрес назначения	Адрес источника	Тип	Данные	FCS
64	48	48	16	368–12000	32

Рис. 2.12. Побитовое представление формата фрейма Ethernet II (DIX)

Примечание

Фрейм Ethernet II иногда называют DIX-фреймом по названию трех компаний первоначально разработавших эту технологию: Digital (Digital Equipment Compапу, позднее приобретенной компанией Compaq), Intel и Xerox.

Во фрейме Ethernet II вместо поля длины используется 16-битное поле типа, предназначенное для сетевых коммуникаций более высокого уровня. Поле данных инкапсулируется без поля-заполнителя и его длина в диапазоне от 368 до 12 000 бит. Переменный размер поля используется для улучшенного обнаружения конфликтов пакетов и оптимизации загрузки сети, чтобы длинные пакеты не занимали сеть в течение слишком большого времени. Последнее поле фрейма Ethernet II – 32-битное поле контрольной суммы фрейма (FCS). С помощью этого поля по тому же алгоритму, как и в традиционном стандарте 802.3, выполняется контроль CRC.

Совет

Во избежание коммуникационных проблем не используйте фреймы Ethernet II и 802.3 для одних и тех же узлов в пределах одной сети.

Как указано в стандарте IEEE 802.3 для коммуникаций на подуровне LLC канального уровня, оба фрейма (802.3 и Ethernet II) могут содержать три необязательных поля между полем длины или типа и полем данных: поле целевой точки доступа к службе (destination service access point, DSAP), поле исходной точки доступа к службе (source service access point, SSAP) и поле управления. Эти поля позволяют Канальному уровню управлять фреймами и взаимодействовать с более высокими уровнями модели OSI. Поля DSAP и SSAP имеют длину 8 бит. Точки доступа к службе (SAP) позволяют сетевому уровню определять, какой сетевой процесс узла назначения должен получать фрейм. Эти точки представляют такие коммуникационные процессы, как OSI, Novell, NetBIOS, TCP IP, BPDU, управление сетями IBM, XNS и другие (описываемые в этой книге). Например, шестнадцатеричное значение EO указывает на Novell SAP, а значение 06 – на SAP стека TCP IP. DSAP указывает точку доступа к службе на целевом узле, который должен принимать фрейм, а SSAP идентифицирует точку доступа к службе передающего узла, который отправляет фрейм. Поле управления определяет функцию (назначение) фрейма (например, указывает на то, что фрейм содержит данные или же код ошибки). Это поле может иметь длину 8 или 16 бит.

Кроме этого, стандарт IEEE 802.3 описывает для LLC реализацию протокола SubNetwork Access Protocol, SNAP (Стандартный протокол доступа к сети), также называемого Ethernet

SNAP. SNAP используется в качестве способа быстрой адаптации протоколов, которые не полностью соответствуют стандартам 802.3 (например, протокола AppleTalk или протокола LAT компании DEC). Когда для подобных протоколов отсутствуют установленные точки SAP, поля DSAP и SSAP содержат шестнадцатеричное значение AA, которое представляет точку SAP для SNAP-фрейма. Кроме этого, поле управления в SNAP-фрейме содержит шестнадцатеричное значение 03. При создании SNAP-фрейма, поле разделителя протоколов помещается сразу же за полем

управления и перед полем данных. Поставщик типа фрейма (например, Apple) идентифицируется первыми тремя байтами поля разделителя протоколов, а тип фрейма Ethernet идентифицируется двумя последними байтами.

Для сетей Ethernet выпускается большое количество оборудования, которое широко поддерживается производителями компьютеров. Одной из причин популярности Ethernet является то, что этот стандарт имеет много решений для реализации высокоскоростных сетей. Например, сети Ethernet с часто той 10 Мбит/с легко модернизировать в сеть Fast Ethernet с частоте 100 Мбит/с, зачастую используя для этого уже установленные сетевые адаптеры и кабельную систему.

17. Сетевые службы и сервисы. Понятие и основные характеристики.

Вообще служба это некоторое программное решение, которое расширяет возможности Вашего ПК. Соответственно сетевые службы это программное обеспечение, которое может наделить Ваш компьютер дополнительными возможностями по работе в сети. Просмотреть службы Вы можете, используя свойства сети. Если на компьютере работает сетевая служба, это значит, что серверное приложение, называемое *демоном*, ожидает подключений к одному или нескольким сетевым портам.

Сетевые службы и протоколы.

Каждый сетевой уровень подчиняется определенному сетевому протоколу, определяющему набор сетевых служб, присущих данному уровню.

Сетевая служба - это набор функций, которые уровень выполняет для выше лежащего уровня.

Протокол уровня определяет структуру данных и формат пакетов для выполнения запрашиваемой сетевой службы.

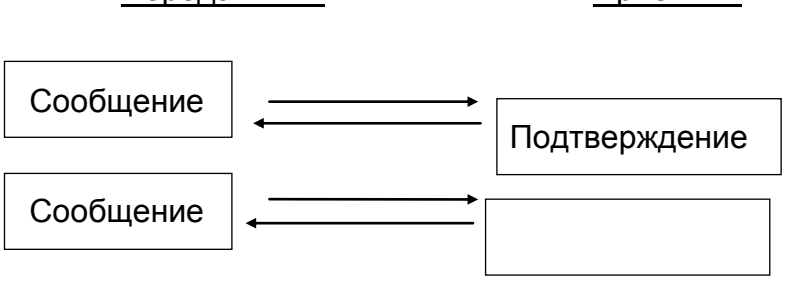
В рамках одной и той же сети для обеспечения одной и той же сетевой службы могут применяться различные методы передачи данных. Эти методы называются режимами (коррекция ошибок, подтверждение получения пакетов и т.д.). Если требуется, чтобы сетевая служба выполняла некоторую функцию, нужно выбрать соответствующий протокол или разработать собственный. Сеть обеспечивает сетевую службу, если требуемый для службы протокол доступен.

Различают сетевые службы, ориентированные и не ориентированные на соединение.

Любое сетевое соединение подразумевает наличие пути между двумя устройствами и наличие самих этих устройств. Двухточечное соединение - это непрерывная цепь между двумя устройствами. Ориентированная на соединение служба устанавливает виртуальное двухточечное соединение (например, телефон). Не ориентированная на соединение служба двухточечного соединения не устанавливает (доставка письма).

Процесс передачи данных по сети не застрахован от ошибок. Поэтому в сетях используется виртуальная коррекция ошибок. Ошибочные пакеты достигают сетевых модулей, но не достигают приложений.

Система контроля ошибок обязана обнаруживать и обрабатывать два вида повреждения данных: искажение и исчезновение. Для этого используются методы CRC и подтверждение получения пакета:



Примеры служб сети Интернет, наиболее широко используемых в настоящее время - это службы электронной почты E-mail и службы Всемирной Паутины - World Wide Web.

- ***DHCP:*** Сетевые службы DHCP разрешают запросы DHCP от ISA-сервера во внутреннюю сеть. Разрешение DHCP-ответов ISA-серверу позволяет ISA-серверу получать доступ к внутренней сети с помощью протоколов ответов и запросов DHCP. Это правило может потребоваться для удаленного доступа или для отслеживания DHCP-доступа. DHCP - открытый промышленный стандарт, который упрощает управление сетями на базе TCP/IP. Каждому хосту (компьютеру), подключенному к сети на базе TCP/IP, должен быть назначен уникальный IP-адрес. Протокол DHCP освобождает сетевых администраторов от необходимости настраивать все компьютеры вручную.

- ***DNS:*** Сетевые службы DNS (разрешение имен с помощью DNS). Разрешают использование ISA-сервером DNS для доступа к выбранным серверам. Разрешают доступ ISA-серверу к ресурсам сети с помощью протокола DNS. С помощью этого элемента политики можно контролировать место разрешения ISA-сервером DNS-запросов, по умолчанию эта установка действует для всех сетей. В случае необходимости можно указать исключения, если у вас присутствует более одного интернет-подключения или вы желаете указать путь разрешения явным образом.

- ***NTP:*** Сетевые службы NTP (Настройка времени). Разрешает NTP-запросы от ISA-сервера доверенным NTP-серверам. Разрешает ISA-серверу доступ ко внутренним ресурсам сети с помощью протокола NTP (UDP). Используется в случае необходимости синхронизации времени с внутренними серверами.

18. Протокол NETBIOS

NetBIOS (Network Basic Input/Output System) — протокол для работы в локальных сетях на персональных ЭВМ типа IBM/PC, разработан в виде интерфейса, который не зависит от фирмы-производителя. Был разработан фирмой Sytek Corporation по заказу IBM в 1983 году. Он включает в себя интерфейс сеансового уровня (англ. NetBIOS interface), в качестве транспортных протоколов использует TCP и UDP.

Особенностью NetBIOS является возможность его работы поверх разных протоколов, самыми распространёнными/известными из которых являются NetBEUI, IPX и стек протоколов TCP/IP; причём если старые версии Windows ориентировались на более лёгкие в реализации и менее ресурсоёмкие NetBEUI и IPX, то современные Windows ориентируются на TCP/IP. При использовании NetBEUI и IPX NetBIOS сам обеспечивает надёжность доставки данных (функциональность SPX не использовалась), а при использовании TCP/IP надёжность доставки обеспечивает TCP, за что удостоился отдельного имени «NBT».

Интерфейс NetBIOS представляет собой стандартный интерфейс разработки приложений (API) для обеспечения сетевых операций ввода/вывода и управления низлежащим транспортным протоколом. Приложения, использующие NetBIOS API интерфейс, могут работать только при наличии протокола, допускающего использование такого интерфейса.

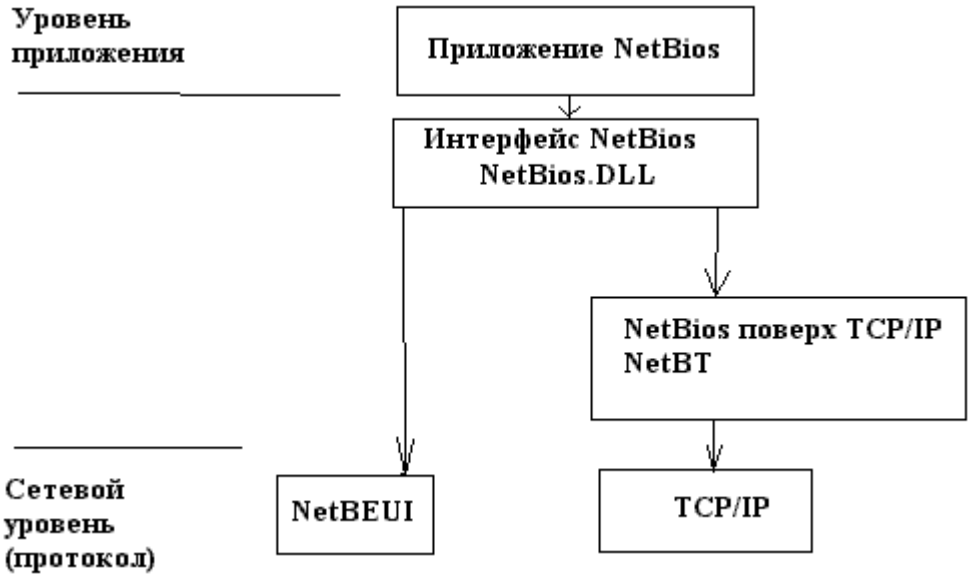
NetBIOS также определяет протокол, функционирующий на сеансовом/транспортном уровнях модели OSI. Этот протокол используется протоколами нижележащих уровней, такими как NBFP (NetBEUI) и NetBT для выполнения сетевых запросов ввода/вывода и операций, описанных в стандартном интерфейсном наборе команд NetBIOS. То есть NetBIOS сам не поддерживает выполнение файловых операций. Эта функция возлагается на протоколы нижележащих уровней, а сам NetBIOS обеспечивает только связь с этими протоколами и NetBIOS API интерфейс.

NetBIOS обеспечивает:
регистрацию и проверку сетевых имен;
установление и разрыв соединений;
связь с гарантированной доставкой информации;
связь с негарантированной доставкой информации;
поддержку управления и мониторинга драйвера и сетевой карты.
NetBios - это сокрашение названия от Network Bios. Это программный интерфейс для приложений клиент - сервер. Устанавливается соединение между клиентом и сервером для перемещения данных в обои стороны. Windows NT поддерживает следующие механизмы установки связи:
Именованные каналы - named pipes
Маилслоты - Mailslot
NetBios
Сокеты Windows
Вызовы удаленных процедур - RPC
Динамический обмен данными по сети - NetDDE
Блоки серверных сообщений - SMB
Распределенная компонентная модель - DCOM

Можно сказать, что NetBios распространил понятие базой операционной системы на сети. Этот интерфейс предоставляет API для разработки сетевых приложений, которые могут

взаимодействовать на межмашинном и межпрограммном уровне. Кроме интерфейса NetBios распространяет и соглашение о именах.

Архитектура приложения с использованием NetBios



На базе интерфейса программирования NetBIOS созданы такие приложения, как Chat входящие в поставку Windows NT до сервера Notes для Windows. Microsoft Mail так же работает с NetBios для уведомления об отправке почты клиента.

NetBios - это программный интерфейс API для создания сетевых приложений. Для его функционирования и использования необходим NetBios.DLL.

19. Протоколы транспортного уровня (TCP, UDP).

Transmission Control Protocol – это протокол, тесно связанный с IP, который используется в аналогичных целях, но на более высоком уровне - транспортном уровне эталонной модели ISO OSI. Часто эти протоколы, по причине их тесной связи, именуют вместе, как TCP/IP. Термин ``TCP/IP'' обычно означает все, что связано с протоколами TCP и IP. Он охватывает целое семейство протоколов, прикладные программы и даже саму сеть. В состав семейства входят протоколы TCP, UDP, ICMP, telnet, FTP и многие другие. Иерархия протоколов семейства TCP/IP показана на Рисунок 6.

TCP/IP - это технология межсетевого взаимодействия, технология internet. Сеть, которая использует технологию internet, называется internet.

Сам протокол TCP занимается проблемой пересылки больших объемов информации, основываясь на возможностях протокола IP. Как это делается? Вполне здраво можно рассмотреть следующую ситуацию. Как можно переслать книгу по почте, если та принимает только письма и ничего более? Очень просто: разодрать ее на страницы и отправить страницы отдельными конвертами. Получатель, руководствуясь номерами страниц, легко сможет книгу восстановить. Этим же простым и естественным методом и пользуется TCP.

TCP делит информацию, которую надо переслать, на несколько частей. Нумерует каждую часть, чтобы позже восстановить порядок. Чтобы пересылать эту нумерацию вместе с данными, он обкладывает каждый кусочек информации своей обложкой - конвертом, который содержит соответствующую информацию. Это и есть

TCP-конверт. Получившийся TCP-пакет помещается в отдельный IP-конверт и получается IP-пакет, с которым сеть уже умеет обращаться.

Получатель (TCP-модуль (процесс)) по получении распаковывает IP-конверты и видит TCP-конверты, распаковывает и их и помещает данные в последовательность частей в соответствующее место. Если чего-то не достает, он требует переслать этот кусочек снова. В конце концов информация собирается в нужном порядке и полностью восстанавливается. Вот теперь этот массив пересылается выше к пользователю (на диск, на экран, на печать).

В действительности, это слегка утрированный взгляд на TCP. В реальности пакеты не только теряются, но и могут искажаться при передаче из-за наличия помех на линиях связи. TCP решает и эту проблему. Для этого он пользуется системой кодов, исправляющих ошибки. Существует целая наука о таких кодировках. Простейшим примером такового служит код с добавлением к каждому пакету контрольной суммы (и к каждому байту бита проверки на четность). При помещении в TCP-конверт вычисляется контрольная сумма, которая записывается в TCP-заголовок. Если при приеме заново вычисленная сумма не совпадает с той, что указана на конверте, значит что-то тут не то, - где-то в пути имели место искажения, так что надо переслать этот пакет по-новой, что и делается.

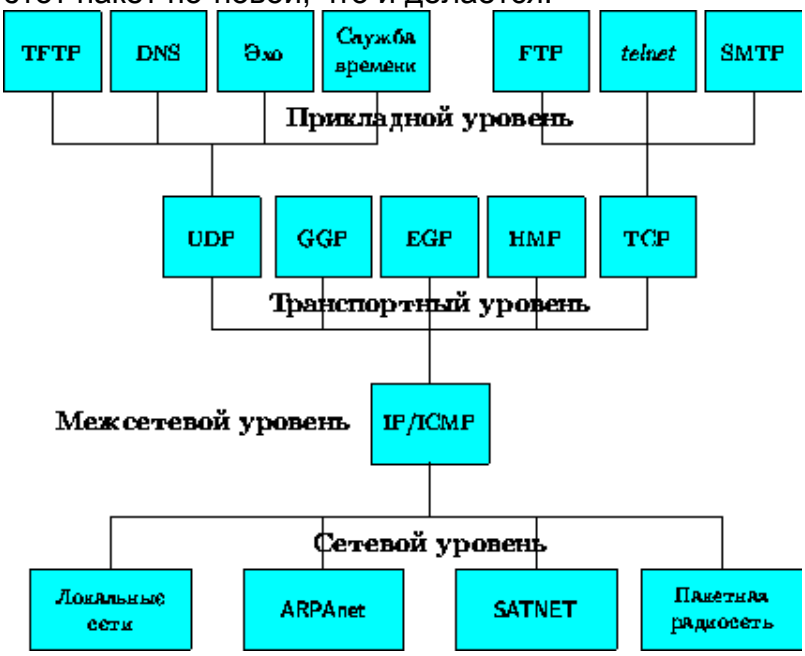


Рисунок 6 Иерархия протоколов

Для ясности и полноты картины, необходимо сделать здесь важное замечание: Модуль TCP разбивает поток байтов на пакеты, не сохраняя при этом границ между записями. Т.е., если один прикладной процесс делает 3 записи в -порт, то совсем не обязательно, что другой прикладной процесс на другом конце виртуального канала получит из своего -порта именно 3 записи, причем именно таких (по разбиению), что были переданы с другого конца. Вся информация будет получена исправно и с сохранением порядка передачи, но она может уже быть разбита по другому и на иное количество частей. Не существует зависимости между числом и размером записываемых сообщений с одной стороны и числом и размером считываемых сообщений с другой стороны. TCP требует, чтобы все отправленные данные были подтверждены принявшей их стороной. Он использует ожидания (таймауты) и повторные передачи для обеспечения надежной доставки. Отправителю разрешается передавать некоторое количество данных, не дожидаясь подтверждения приема ранее отправленных данных. Таким образом, между отправленными и подтвержденными данными существует окно уже отправленных, но еще не подтвержденных данных. Количество байт, которое можно передавать без подтверждения, называет-

ся размером окна. Как правило, размер окна устанавливается в стартовых файлах сетевого программного обеспечения. Так как TCP-канал является , т.е. данные могут одновременно передаваться в обоих направлениях, то подтверждения для данных, идущих в одном направлении, могут передаваться вместе с данными, идущими в противоположном направлении. Приемники на обеих сторонах виртуального канала выполняют управление потоком передаваемых данных для того, чтобы не допускать переполнения буферов.

Таким образом, протокол TCP обеспечивает гарантированную доставку с установлением логического соединения в виде байтовых потоков. Он освобождает прикладные процессы от необходимости использовать ожидания и повторные передачи для обеспечения надежности. Наиболее типичными прикладными процессами, использующими TCP, являются ftp и telnet. Кроме того, TCP использует система X-Windows (стандартный многооконный графический интерфейс с пользователем), ``r-команды".

Большие возможности TCP даются не бесплатно, реализация TCP требует большой производительности процессора и большой пропускной способности сети. Когда прикладной процесс начинает использовать TCP, то начинают общаться модуль TCP на машине пользователя и модуль на машине сервера. Эти два оконечных модуля TCP поддерживают информацию о состоянии соединения - виртуального канала. Этот виртуальный канал потребляет ресурсы обоих оконечных модулей TCP. Канал этот, как уже указывалось, является дуплексным. Один прикладной процесс пишет данные в TCP-порт, откуда они модулями соответствующих уровней по цепочке передаются по сети и выдаются в TCP-порт на другом конце канала, и другой прикладной процесс читает их отсюда - из своего TCP-порта. эмулирует (создает видимость) выделенную линию связи двух пользователей. Гарантирует неизменность передаваемой информации. Что входит на одном конце, выйдет с другого. Хотя в действительности никакая прямая линия отправителю и получателю в безраздельное владение не выделяется (другие пользователи могут пользовать те же узлы и каналы связи в сети в промежутках между пакетами этих), но извне это, практически, именно так и выглядит.

Как бы хорошо это не звучало, но это не панацея. Как уже отмечалось, установка TCP-виртуального канала связи требует больших расходов на инициирование и поддержание соединения и приводит к задержкам передачи. Если вся эта суета - излишество, лучше обойтись без нее. Если все данные, предназначенные для пересылки, умещаются в одном пакете, и если вас не особенно заботит надежность доставки (? - читайте дальше, - поймете), то можно обойтись без TCP.

Если для приложения контроль качества передачи данных по сети имеет значение, то в этом случае используется протокол TCP. Этот протокол еще называют надежным, ориентированным на соединение и потокоориентированным протоколом. Прежде чем обсудить эти свойства протокола, рассмотрим формат передаваемой по сети датаграммы (Рисунок 7). Согласно этой структуре, в TCP, как и в UDP, имеются порты. Первые 256 портов закреплены за WKS, порты от 256 до 1024 закреплены за Unix-сервисами, а остальные можно использовать по своему усмотрению. В поле Sequence Number определен номер пакета в последовательности пакетов, которая составляет все сообщение, за тем идет поле подтверждения Asknowledgment Number и другая управляющая информация.

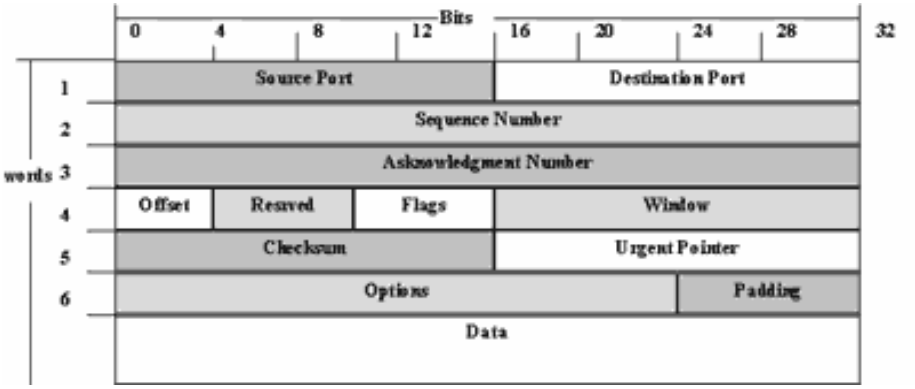


Рисунок 7 Структура пакета TCP

Надежность TCP заключается в том, что источник данных повторяет их посылку, если только не получит в определенный промежуток времени от адресата подтверждение об их успешном получении. Этот механизм называется Positive Asknowledgement with Retransmission (PAR). Как мы ранее определили, единица пересылки (пакет данных, сообщение и т.п.) в терминах TCP носит название сегмента. В заголовке TCP существует поле контррольной суммы. Если при пересылке данные повреждены, то по контрольной сумме модуль, вычленяющий TCP-сегменты из пакетов IP, может определить это. Поврежденный пакет уничтожается, а источнику ничего не посылается. Если данные не были повреждены, то они пропускаются на сборку сообщения приложения, а источнику отправляется подтверждение.

Ориентация на соединение определяется тем, что прежде чем отправить сегмент с данными, модули TCP источника и получателя обмениваются управляющей информацией. Такой обмен называется handshake (буквально "рукопожатие"). В TCP используется трехфазный hand-shake:

1. Источник устанавливает соединение с получателем, посылая ему пакет с флагом "синхронизации последовательности номеров" (Synchronize Sequence Numbers - SYN). Номер в последовательности определяет номер пакета в сообщении приложения. Это не обязательно должен быть 0 или единица. Но все остальные номера будут использовать его в качестве базы, что позволит собрать пакеты в правильном порядке;

2. Получатель отвечает номером в поле подтверждения получения SYN, который соответствует установленному источником номеру. Кроме этого, в поле "номер в последовательности" может также сообщаться номер, который запрашивался источником;

3. Источник подтверждает, что принял сегмент получателя и отправляет первую порцию данных.

Графически этот процесс представлен на **Рисунок 8**

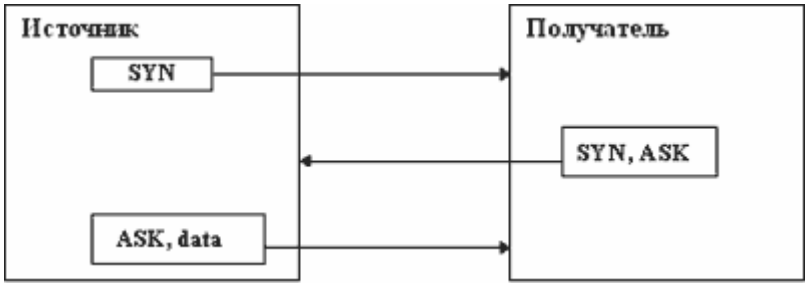


Рисунок 8 Установка соединения TCP

После установки соединения источник посылает данные получателю и ждет от него подтверждений о их получении, затем снова посылает данные и т.д., пока сообщение не закончится. Заканчивается сообщение, когда в поле флагов выставляется бит FIN, что означает "нет больше данных".

Потоковый характер протокола определяется тем, что SYN определяет стартовый номер для отсчета переданных байтов, а не пакетов. Это значит, что если SYN был установлен в 0, и было передано 200 байтов, то номер, установленный в следующем пакете будет равен 201, а не 2.

Понятно, что потоковый характер протокола и требование подтверждения получения данных порождают проблему скорости передачи данных. Для ее решения используется "окно" - поле - window. Идея применения window достаточно проста: передавать данные не дожидаясь подтверждения об их получения. Это значит, что источник предает некоторое количество данных равное window без ожидания подтверждения об их приеме, и после этого останавливает передачу и ждет подтверждения. Если он получит подтверждение только на часть переданных данных, то он начнет передачу новой порции с номера, следующего за подтвержденным. Графически это изображено на рисунке Рисунок 9.

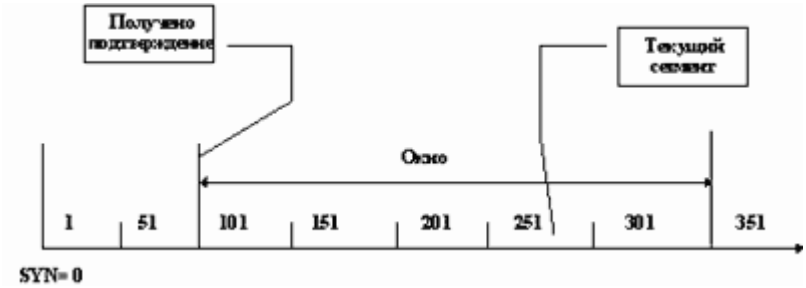


Рисунок 9 Механизм передачи данных по TCP

В данном примере окно установлено в 250 байтов шириной. Это означает, что текущий сегмент - сегмент со смещением относительно SYN, равном 250 байтам. Однако, после передачи всего окна модуль TCP источника получил подтверждение на получение только первых 100 байтов. Следовательно, передача будет начата со 101 байта, а не с 251.

Таким образом, мы рассмотрели все основные свойства протокола TCP. Осталось только назвать наиболее известные приложения, которые использует TCP для обмена данными. Это в первую очередь TELNET и FTP, а также протокол HTTP, который является сердцем World Wide Web.

Прервем немного разговор о протоколах и обратим свое внимание на такую важнейшую компоненту всей системы TCP/IP как IP-адреса.

UDP

Имеется другой стандартный протокол транспортного уровня, который не отягощен такими накладными расходами. Этот протокол называется **UDP – User Datagram Protocol** - протокол пользовательских дейтаграмм. Он используется вме-

сто TCP. Здесь данные помещаются не в TCP, а в UDP-конверт, который также помещается в IP-конверт. Этот протокол реализует дейтаграммный способ передачи данных.

Дейтаграмма - это пакет, передаваемый через сеть независимо от других пакетов без установления логического соединения и подтверждения приема. Дейтаграмма - совершенно самостоятельный пакет, поскольку сама содержит всю необходимую для ее передачи информацию. Ее передача происходит безо всякого предварения и подготовки. Дейтаграммы, сами по себе, не содержат средств обнаружения и исправления ошибок передачи, поэтому при передаче данных с их помощью следует принимать меры по обеспечению надежности пересылки информации. Методы организации надежности могут быть самыми разными, обычно же используется метод подтверждения приема посылкой эхо-отклика при получении каждого пакета с дейтаграммой.

UDP проще TCP, поскольку он не заботится о возможной пропаже данных, пакетов, о сохранении правильного порядка данных и т.д. UDP используется для клиентов, которые посылают только короткие сообщения и могут просто заново послать сообщение, если отклик подтверждения не придет достаточно быстро. Предположим, что вы пишите программу, которая просматривает базу данных с телефонными номерами где-нибудь в другом месте сети. Совершенно незачем устанавливать TCP связь, чтобы передать 33 или около того символов в каждом направлении. Вы можете просто уложить имя в UDP-пакет, запаковать это в IP-пакет и послать. На другом конце прикладная программа получит пакет, прочитает имя, посмотрит телефонный номер, положит его в другой UDP-пакет и отправит обратно. Что произойдет, если пакет по пути потеряется? Ваша программа тогда должна действовать так: если она ждет ответа слишком долго и становится ясно, что пакет затерялся, она просто повторяет запрос, т.е. посылает еще раз то же послание. Так обеспечивается надежность передачи при использовании протокола UDP.

В отличие от TCP, данные, отправляемые прикладным процессом через модуль UDP, достигают места назначения как единое целое. Например, если процесс-отправитель производит 3 записи в UDP-порт, то процесс-получатель должен будет сделать 3 чтения. Размер каждого записанного сообщения будет совпадать с размером соответствующего прочитанного. Протокол UDP сохраняет границы сообщений, определяемые прикладным процессом. Он никогда не объединяет несколько сообщений в одно целое и не делит одно сообщение на части.

Протокол UDP - это один из двух протоколов транспортного уровня, которые используются в стеке протоколов TCP/IP. UDP позволяет прикладной программе передавать свои сообщения по сети с минимальными издержками, связанными с преобразованием протоколов уровня приложения в протокол IP. Однако при этом, прикладная программа сама должна заботиться о подтверждении того, что сообщение доставлено по месту назначения. Заголовок UDP-датаграммы (сообщения) имеет вид, показанный на Рисунок 10.

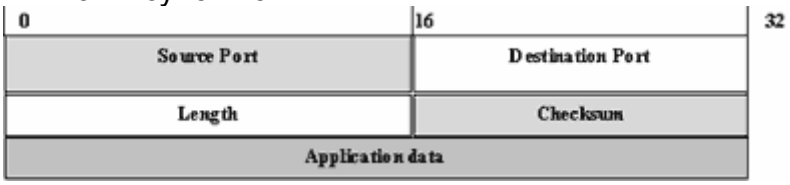


Рисунок 10 Заголовок UDP-датаграммы

Порты в заголовке определяют протокол UDP как мультиплексор, который позволяет собирать сообщения от приложений и отправлять их на уровень протоколов. При этом приложение использует определенный порт. Взаимодействующие через сеть приложения могут использовать разные порты, что и отражает заголовок пакета. Всего можно определить 216 разных портов. Первые 256 портов закреплены за,

так называемыми "well known services", к которым относятся, например, 53 порт UDP, который закреплен за сервисом DNS.

Поле Length определяет общую длину сообщения. Поле Checksum служит для контроля целостности данных. Приложение, которое использует протокол UDP должно само заботится о целостности данных, анализируя поля Checksum и Length. Кроме этого, при обмене данными по UDP прикладная программа сама должна заботится о контроле доставки данных адресату. Обычно это достигается за счет обмена подтверждениями о доставке между прикладными программами.

Наиболее известными сервисами, основанными на UDP, является служба доменных имен BIND и распределенная файловая система NFS. Если возвратиться к примеру traceroute, то в этой программе также используется транспорт UDP. Собственно, именно сообщение UDP и засылается в сеть, но при этом используется такой порт, который не имеет обслуживания, поэтому и порождается ICMP-пакет, который и детектирует отсутствие сервиса на принимающей машине, когда пакет наконец достигает машину-адресата.

20. Понятие “socket”. Службы, вызовы, принципы работы.

Программный интерфейс Sockets впервые появился в BSD Unix 4.1c в 1982 г. Основная цель создания данного интерфейса заключалась в возможно более полном сокрытии деталей конкретной реализации сетевых примитивов. Механизм, предоставляемый с помощью Sockets, не зависит от того, действительно ли устанавливается сетевое соединение или же обмен информацией происходит на локальной машине.

Интерфейс Sockets реализован практически во всех версиях операционной системы Unix. Как программный объект, socket является абстракцией и представляет собой конечную точку коммуникационной структуры. Каждый socket связан с коммуникационным доменом, т. е. с группой протоколов, использующих одинаковые транспорт и правила трансляции имен. Наиболее известен Internet-домен, в который входят протоколы TCP, UDP, IP и т. д.

Работа с Sockets во многом напоминает файловые операции, часто для передачи данных можно использовать обычные файловые операции. В приведенном на рис. 2 фрагменте программы представлены оба подхода к передаче данных через интерфейс Sockets: с помощью обычного системного вызова write() и с помощью специфического для socket вызова sendto(). Следует обратить внимание на функцию (макроподстановку) htons, которая предназначена для преобразования информации от представления, принятого на данном компьютере, к представлению, стандартному для передачи информации по сети. Необходимость ее использования, равно как и обратной функции ntohs, связана с различным представлением данных для различных компьютеров. Порядок байтов в слове зависит от архитектуры процессора. В зависимости от конкретного процессора указанные функции могут выполнять преобразования информации, а могут и ничего не делать.

На рис. 2 приведен пример программы, в которой есть все основные операции для установления соединения с удаленным компьютером с помощью механизма Sockets. В тексте программы расставлены цифровые метки в виде комментариев. Они обозначают начало блоков операций, описанных ниже. Нумерация этих блоков будет использована также при описании других программных интерфейсов.

Итак, рассмотрим последовательность вызовов для клиентской части.

1. Создание socket, относящегося в данном примере к классу сетевых (параметр AF_INET) и типу потоковых (SOCK_STREAM)3. Для протокола IP это означает socket для TCP.

2. Установление соединения (connect) с машиной, которая имеет имя, переданное программе в виде параметра argv[1], преобразованное и сохраненное в поле serv_name.sin_addr; порт для соединения задан константой PORT_UNIQUE.

3. Обмен данными может осуществляться с помощь функции write (для файловых операций) или send.

4. Закрытие socket.

Для серверной части после создания socket добавляется операция по привязке (bind) объекта к конкретному порту.

21. Представление FCP по уровням модели OSI (физический уровень, кодирование передаваемой информации, контроль канала передачи, сервис передачи данных, FC-4).

среда распространения

Любое устройство (компьютер, сервер, принтер, накопитель и т.д.), имеющее возможность обмениваться данными с использованием технологии Fibre Channel, называется N_порт (Node port) или просто узел. В настоящее время межузловой обмен происходит по полнодуплексным последовательным соединениям со скоростью 1.0625 GBit/s.

Множество связанных между собой N_портов Fibre Channel образуют среду распространения сигнала. Во избежание смешения с устоявшимися понятиями разработчики данной технологии решили не использовать для ее обозначения слово "network", и предпочли ввести новый термин "fabric". Учитывая, что нам до сих пор ни разу не попадались удачные примеры перевода этого слова на русский язык, предлагаем хотя бы в рамках данной статьи называть такую среду распространения "решеткой".

Понятие решетки аппаратно независимо и не рассматривает топологию физических межсоединений между ее отдельными узлами. Единственное ограничение заключается в максимально допустимом размере адресного пространства 2^{24}, т.е. более 16 миллионов узлов (эквивалентно сети класса A).

архитектурная модель FC-AL

Архитектурная модель Fibre Channel в деталях описывает параметры соединений и протоколы обмена между отдельными узлами решетки. Примерно как и в случае сетевых протоколов, эта модель может быть представлена в виде многослойного стека функциональных уровней, хотя и без непосредственного соответствия с OSI. Пять функциональных уровней архитектуры Fibre Channel определяют физический интерфейс, схему кодировки, сигнальный протокол, общие процедуры и связь с приложениями. Нумерация идет от самого низкого аппаратного уровня FC-0, отвечающего за параметры физическо-

го соединения до верхнего программного FC-4, взаимодействующего с приложениями более высокого уровня.

среда передачи и физические интерфейсы (fc-0)

На данном уровне задаются физические параметры полнодуплексного последовательного соединения между портами. В качестве среды передачи может быть использована витая пара, коаксиальный или твинаксиальный кабели, а также многомодовое или одномодовое волокно. Источниками света могут служить как мощные длинноволновые OFC лазеры, так и менее дорогие non-OFC. Для снижения задержек и уменьшения электрических и температурных перепадов по соединению идет постоянная и равномерная передача и прием сигнала. Кроме того, в целях предотвращения низкочастотных токов структура передаваемого сигнала должна быть сбалансирована постоянным чередованием 0 и 1 вне зависимости от присутствия в нем полезных данных, что упрощает точное распознавание сигнала и снижает количество ошибок. Упрощенно говоря, в FC-0 определяется способ передачи полученных с более высокого уровня бинарных последовательностей. Нелишне также отметить, что именно данный уровень позволяет изменять скорости передачи от 250Mbit/s до 8Gbit/s, не затрагивая остальные более высокие уровни.

Протокол передачи (fc-1)

Протокол передачи определяет, как "вплести" данные в нижележащие сигналы FC-0, как установить соединения между портами и как, в случае необходимости, исправить обнаруженную ошибку. На этом уровне с помощью кодировки IBM 8b/10b 8-битовые порции данных преобразуются в 10-битовые сбалансированные по количеству 0 и 1 последовательности, что требуется для корректной работы FC-0. Для этого каждый планируемый к передаче байт (любой из 256 символов ASCII) преобразуется в четыре возможных комбинации для 10-битового представления, после чего выбираются две наиболее сбалансированные. Здесь действуют два простых правила - не менее 4 нулей и единиц в 10-битовой последовательности и не более четырех 0 или 1 подряд. В итоге, из двух предложенных ему на выбор 10-битовых последовательностей уровень FC-0 передает ту, первый символ которой отличается от последнего символа предыдущей. Таким образом, кодировка 8b/10b выполняет функцию НЧ фильтра, не пропуская низкочастотную составляющую и существенно облегчая работу приемника.

На этом уровне также используется символ K28.5 (только не спрашивайте нас, что означает K28.5, лучше сразу звоните на IBM :-). Эта 10-битовая последовательность не может быть получена из символов ASCII путем преобразования 8b/10b, поэтому может смело применяться в качестве служебной. Разумеется, для соответствия объявленным правилам передачи и этот символ существует как в виде позитива (1100000101), так и негатива (0011111010). В соответствии с теми же правилами последовательность из пяти 0 или 1 не может встретиться при передаче данных и однозначно определяется, как служебный сигнал.

В результате применяемых на уровнях FC-0/1 алгоритмов вероятность возникновения ошибки на переданный бит (BER - Bit Error Rate) составляет ничтожную величину 10-12, что на три порядка лучше, чем для применяемых в SCSI или Ethernet асинхронных способов передачи.

Сигнальный протокол (fc-2)

Вот мы и дошли до самого интересного, составляющего основное ядро Fibre Channel. Сигнальный протокол определяет иерархическую структуру посылок для установления связей между работающими через FC-AL приложениями. Основными объектами этого уровня являются слова (words), кадры (frames), пакеты (sequences) и обмены (exchanges).

Базовым элементом и минимальной единицей передачи является слово, но пересылка данных между узлами FC-AL требует помещения слов в некий контейнер. Ничего удивительного, что такой контейнер назван кадром.

Один или несколько последовательных кадров, несущих помещенную в них связанную информацию в виде файла данных, графики, программы или же IP-пакета, называются пакетом, который представляет из себя однонаправленную посылку от передающего узла принимающему.

Набор пакетов, которыми обмениваются узлы для обслуживания работающего через них приложения, называется обменом. Обмен, понимаемый как диалог между двумя приложениями высокого уровня, является двунаправленным, и после своего начала может оставаться открытым сколь угодно долго. Такая синтаксическая конструкция является отличительной чертой Fibre Channel и позволяет ему поддерживать огромное количество протоколов одновременно.
Здесь нужно обратить внимание, что сама решетка Fiber Channel умеет работает исключительно с кадрами, доставляя их между отправляющим и получающим узлами. Она даже не подозревает о более сложных синтаксических конструкциях в виде пакетов или обменов, которые создаются и разбираются самими узлами.

Слова (words)

Передача в Fibre Channel идет строго 4-байтовыми словами, т.е. 40-битовыми последовательностями на уровне FC-0. Слова идут слитно и постоянно вне зависимости от наличия информации для передачи, когда передающий порт генерирует слова-пустышки (IDLE) в соответствии с требованиями уровня FC-0.

Для пересылки данных все байты представлены в виде символов ASCI. Если первый байт слова, вернее даже, его первые 10 бит на уровне FC-1 замещаются на служебный символ K28.5, то оно распознается, как служебное (ordered set) и, в зависимости от содержимого трех остальных байт, приобретает различную смысловую нагрузку. Примерами служебных слов могут быть IDLE (постоянная передача, когда все равно больше нечем заняться), ARB (запрос на арбитраж в петле), SOF (начало кадра) и EOF (конец кадра).

Кадры (frames)

Служат "транспортными контейнерами" для пересылок между отдельными узлами решетки. Начало кадра определяется служебным словом SOF (Start Of Frame), структуру которого мы рассматривали ранее. Непосредственно за SOF располагаются шесть слов заголовка и от 0 до 528 смысловых слов. Кадр завершается контрольным словом CRC (Cyclic Redundancy Check) и EOF (End Of Frame). Другими словами, размер каждого кадра может варьировать от 9 до 537 слов, что позволяет его гибко подстраивать под объем передаваемой информации.

Следующее сразу за SOF первое слово заголовка (header word 0) содержит адрес маршрутизации по решетке Fibre Channel. Остальные пять слов заголовка содержат уникальные идентификаторы последовательностей (header word 1), обменов (header word 2), определяют номер кадра в этих конструкциях (header word 3), относительное смещение в оперативной памяти (header word 4) и тип сообщений (header word 5) — SCSI, IP, AV, VI etc.

Далее размещаются смысловые слова, ради которых все это и затевалось. Их количество в кадре может варьировать от 0 до 528 и определяется передающим портом с учетом собственных возможностей и возможностей остальных портов, информацию о чем он получает во время процедуры подключения (log-in).

Служебное слово CRC является всегда предпоследним в кадре и служит для проверки правильности передачи заголовка и смысловых слов на основе содержащихся в нем контрольных 4 байт (32 бит).

Как уже все догадались, кадр закрывается с помощью слова EOF. И только после получения EOF порт опознает предыдущее слово, как CRC, ведь в нем все 4 байта являются контрольными и служебному символу K28.5 попросту не хватило места.

После столь детального разбора структуры кадра пришло время рассказать про одну маленькую деталь, о которой обычно умалчивают разработчики Fibre Channel в публикациях, рассчитанных на массовую аудиторию потенциальных пользователей. Ее некоторая скан-

дальность заключается в том, что после завершения кадра N_порт обязан также передать не меньше 6 слов IDLE.

В основе требования передачи шести слов IDLE между кадрами заложена та же самая идея, что и при выборе кодировки 8b/10b - пожертвовать некоторой частью полосы канала для получения простого, недорогого и вместе с тем надежного механизма передачи. Кодировка 8b/10b "съедает" 20% пропускной способности канала в обмен на простой и надежный механизм распознавания слов. Шесть слов IDLE между кадрами делают примерно то же самое для обработки самих кадров. Дело в том, что в процессе передвижения кадра по решетке все встречающиеся ему на пути узлы имеют слегка отличающиеся собственные опорные частоты. В итоге, узел с несколько меньшей собственной частотой посылает кадры несколько медленнее, чем принимает, поэтому во избежание "наползания" кадров друг на друга он просто время от времени выкидывает слова IDLE между ними. С той же целью более "быстрый" узел также время от времени может вставлять недостающие IDLE, чтобы "не наступать на пятки" своему соседу.

Надеемся, что читатели специализированного издания поймут всю красоту такого подхода, когда можно не предъявлять высокие требования к кварцевым резонаторам разных производителей, к тому же работающим в разных температурных условиях.

Здесь сразу уместен вопрос о реальной пропускной способности Fibre Channel. Уже на ранних этапах разработки стандарта стало очевидным, что производители и пользователи вкладывают в это определение различный смысл, поэтому во избежание путаницы было принято соломоново решение. В настоящее время производители оборудования должны указывать аппаратную скорость компонентов в мегабитах или гигабитах в секунду, в то время как пользователи должны руководствоваться реальной скоростью передачи данных, измеряемой мегабайтами в секунду.

Если вспомнить о 20% избыточности 8b/10b, то в данном случае 1Gbit/s = 100MBytes/s. А чтобы иметь возможность обеспечить поток пользовательских данных 100MBytes/s с учетом избыточности CRC, адресных заголовков и других служебных слов, сейчас на уровне FC-0 используется скорость аппаратной передачи 1.0625Gbit/s.

Таким образом, учитывая полнодуплексную реализацию Fibre Channel, можно говорить о представлении пользователю 200MBytes/s в случае симметичности потоков данных, т.е. в контексте технологии FC-AL 1.0625Gbit/s = 200MBytes/s.

Пакеты (sequences)

Синтаксическая конструкция пакетов FC-AL определяется как серия из одного или нескольких кадров, несущих отдельные порции единого блока информации. К примеру, SCSI команда вполне помещается в один кадр и, в данном случае, рассматривается, как пакет. Совершенно такая же ситуация и с 2GB блоком данных, только пакет в этом случае будет состоять из 960 млн. кадров. Блок данных всегда передается последовательно, что определяется протоколами более высокого уровня. Тем не менее, если при движении по решетке некоторые кадры будут задержаны, принимающий порт способен корректно собрать принятые кадры в блок данных на основании содержащихся в заголовке SEQ_ID (идентификатор последовательности) и SEQ_CNT (порядковый номер кадра в последовательности).

Обмены (exchanges)

Каждое взаимодействие между приложениями через Fibre Channel происходит в контексте обмена. Каждый обмен имеет инициатора (originator) и ответчика (responder). Для начала обмена инциатор посылает первый кадр первого пакета обмена ответчику. Содержимое кадра может составлять, к примеру, SCSI команду. В заголовке этого кадра инициатор присваивает значение OX_ID (exchange originator identification). После этого все кадры данного обмена будут возвращаться от ответчика с этим же OX_ID, что позволит инициатору получать контекстную информацию о приложениях и протоколах более высокого уровня из собственной таблицы соответствий. Одновременно с этим ответчик приваивает собственное значение RX_ID (exchange responder identification) в первом кадре своего первого пакета в

пределах данного обмена. После получения этого кадра инициатором все дальнейшие кадры содержат уникальные идентификаторы сторон в контексте данного обмена, что позволяет точно установить принадлежность кадров при нескольких одновременных обменах. К примеру, кадры пакетов обмена SCSI командами могут приходить вперемешку с кадрами пакетов обмена IP, но порт сможет рассортировать их "на лету", отделив мух от котлет. Когда обмен завершен, соответствующие значения OX_ID и RX_ID освобождаются для использования в будущих обменах. Таким образом N_порт FC-AL может рассматриваться, как многопротокольный маршрутизатор.

Каждый порт способен начать и поддерживать до 64 тыс. конкурентных обменов. Одновременно с этим он способен отвечать еще на 64 тыс. обменов с их поддержкой.

Оригинальность синтаксической структуры Fiber Channel, поддерживаемая словами 1-5 заголовка, разительно отличает его от остальных протоколов передачи. Во-первых, порт имеет возможность конкурентной поддержки различных протоколов высокого уровня. Во-вторых, управление типами передачи выполняется на аппаратном уровне с минимальными микросекундными задержками. И, самое главное, все это происходит без участия системной шины, центрального процессора и операционной системы, т.е. не создавая потенциально узкие места.

Общие процедуры (fc-3)

Этот уровень зарезервирован под описание общих процедур при наличии двух или более N_портов в хосте. Одним из примеров такой процедуры является образование "группы захвата" (hunt group), когда два или более N_портов объединяются под единым адресом, что позволяет увеличить пропускную способность канала от порта до Fibre Channel решетки.

Отображение протоколов (fc-4)

Как и все предыдущие уровни, FC-4 также является чисто аппаратным и отвечает за преобразование различных протоколов в сигнальный протокол FC-AL и обратно. На основании заголовка пришедшего кадра содержащиеся в нем данные преобразуются и помещаются в область оперативной памяти, выделенную для приложения. Понятно, что вряд-ли эта функция может быть реализована программно при скоростях обмена 200Mbytes/s, вследствие чего на этом уровне хранится аппаратный набор логических матриц (profiles), определяющих бинарное соответствие FC-AL и наследуемых протоколов высокого уровня.

По нашему глубокому убеждению, именно качество стандартизации логических матриц данного уровня будет определять коммерческий успех технологии Fibre Channel в целом. Дело в том, что преобразование одного протокола в другой может производиться различными методами и на первых порах производителям не всегда удавалось договориться, чей же метод эффективнее и быстрее, особенно когда каждый из них уже потратил деньги на разработку собственного набора микросхем. Тем не менее, на сегодняшний день методы преобразования SCSI (для жестких дисков) и IP уже определены и приняты в виде стандарта. В ближайшее время будет завершена работа по принятию таких же единых методов преобразования для недисковых устройств SCSI, а также FDDI, Token Ring, ATM, AV, VI, IBM SBCCS, HIPPI и многих других. После того, как метод преобразования конкретного протокола принимается, его алгоритм становится доступным для любого производителя, что делает бессмысленным использование собственного метода и гарантирует совместимость различных устройств.

22. Понятие о GRID технологиях.

Технология Грид (Grid) используется для создания географически распределенной вычислительной инфраструктуры, объединяющей ресурсы различных типов с коллективным доступом к этим ресурсам в рамках виртуальных организаций, состоящих из предприятий и специалистов, совместно использующих эти общие ресурсы.

Идейной основой технологии Грид является объединение ресурсов путем создания компьютерной инфраструктуры нового типа, обеспечивающей глобальную интеграцию информационных и вычислительных ресурсов на основе сетевых технологий и специального программного обеспечения промежуточного уровня (между базовым и прикладным ПО), а также набора стандартизованных служб для обеспечения надежного совместного доступа к географически распределенным информационным и вычислительным ресурсам: отдельным компьютерам, кластерам, хранилищам информации и сетям.

С самых общих позиций эта технология характеризуется простым набором критериев:

- координация использования ресурсов при отсутствии централизованного управления этими ресурсами;
- использование стандартных, открытых, универсальных протоколов и интерфейсов;
- обеспечение высококачественного обслуживания пользователей.

Грид является технологией обеспечения гибкого, безопасного и скоординированного общего доступа к ресурсам. При этом слово «ресурс» понимается в очень широком смысле, т.е. ресурсом может быть аппаратура (жесткие диски, процессоры), а также системное и прикладное ПО (библиотеки, приложения).

В терминологии Грид совокупность людей и организаций, решающих совместно ту или иную общую задачу и предоставляющих друг другу свои ресурсы, называется виртуальной организацией (ВО). Например, виртуальной организацией может быть совокупность всех людей, участвующих в какой-либо научной коллаборации.

Рисунок. Уровни арх-ры протоколов Грид и их соот-е уровням арх-ры протоколов Интернет

Ниже на рисунке приведен базовый уровень.

Рис. Ресурсы Грид

Уровень связи (Connectivity Layer) определяет коммуникационные протоколы и протоколы аутентификации.

Протоколы уровня связи должны обеспечивать надежный транспорт и маршрутизацию сообщений, а также присвоение имен объектам сети. Сейчас протоколы уровня связи в Грид-системах предполагают использование только стека протоколов TCP/IP, в частности: на сетевом уровне – IP и ICMP, транспортном уровне – TCP, UDP, на прикладном уровне – HTTP, FTP, DNS, RSVP.

Коммуникационные протоколы обеспечивают обмен данными между компонентами базового уровня.

Ресурсный уровень (Resource Layer) построен над протоколами коммуникации и аутентификации уровня связи архитектуры Грид. Ресурсный уровень реализует протоколы, обеспечивающие выполнение следующих функций:

- согласование политик безопасности использования ресурса;
- процедура инициации ресурса;
- мониторинг состояния ресурса;
- контроль над ресурсом;
- учет использования ресурса

Коллективный уровень (Collective Layer) отвечает за глобальную интеграцию различных наборов ресурсов, в отличие от ресурсного уровня, сфокусир-го на работе с отдельно взятыми ресурсами. В кол.уровне различают общие и специфич-е (для приложений) протоколы. К общим протоколам относятся протоколы обнаружения и выделения ресурсов, системы мониторинга и авторизации сообществ. Специфические протоколы создаются для различных приложений Грид, (напр, протокол архивации распределенных данных или протоколы управления задачами сохранения состояния и т.п.).

Компоненты колл.уровня предлагают огромное разнообразие методов совместного использования ресурсов. Ниже приведены функции и сервисы, реализуемые в протоколах данного уровня:

- сервисы каталогов позволяют виртуальным организациям обнаруживать свободные ресурсы, выполнять запросы по именам и атрибутам ресурсов, таким как тип и загрузка;
- сервисы совместного выделения, планирования и распределения ресурсов обеспечивают выделение одного или более ресурсов для определенной цели, а также планирование выполняемых на ресурсах задач;
- сервисы мониторинга и диагностики отслеживают аварии, атаки и перегрузку.
- сервисы дублирования (репликации) данных координируют использование ресурсов памяти в рамках виртуальных организаций, обеспечивая повышение скорости доступа к данным в соответствии с выбранными метриками, такими как время ответа, надежность, стоимость и т.п.;
- сервисы управления рабочей загрузкой применяются для описания и управления многошаговыми, асинхронными, многокомпонентными заданиями;
- службы авторизации сообществ способствуют улучшению правил доступа к разделяемым ресурсам, а также определяют возможности использования ресурсов сообщества. Позволяют формировать политики доступа на основе информации о ресурсах, протоколах управления ресурсами и протоколах безопасности связывающего уровня;
- службы учета и оплаты обеспечивают сбор информации об использовании ресурсов для контроля обращений пользователей;
- сервисы координации поддерживают обмен информацией в потенциально большом сообществе пользователей.

Прикладной уровень (Application Layer) описывает польз-кие приложения, работающие в среде виртуальной организации. Приложения функционируют, используя сервисы, определенные на нижележащих уровнях. На каждом из уровней имеются протоколы, обеспечивающие доступ к необходимым службам, а также прикладные программные интерфейсы (Application Programming Interface – API), соотв. данным протоколам.

Сущ-ют инструм-ные средства для управления GRID, которые позволяют управлять ресурсами (выделять, отнимать, перераспределять), процессами, управлять способами представления информации, защитой системы, библиотеками прикладных программ. Пример - Globus Toolkit/

23. Сетевые антивирусные средства. Классификация, приницпы работы.

Антивирусная программа (антивирус) — любая программа для обнаружения компьютерных вирусов, а также нежелательных (считающихся вредоносными) программ вообще и восстановления зараженных (модифицированных) такими программами файлов, а также для профилактики — предотвращения заражения (модификации) файлов или операционной системы вредоносным кодом.Содержание [убрать]

Целевые платформы антивирусного ПО

На данный момент антивирусное программное обеспечение разрабатывается в основном для ОС семейства Windows от компании Microsoft, что вызвано большим количеством вредоносных программ именно под эту платформу (а это вызвано большой популярностью этой ОС, также как и большим количеством средств разработки, в том числе бесплатных и даже "инструкций по написанию вирусов"). Однако в настоящий момент на рынок выходят продукты и под другие платформы настольных компьютеров, такие как – Linux и Mac OS X, это вызвано началом распространения вредоносных программ и под эти платформы, хотя обычно Юникс-подобные системы всегда славились своей надежностью. Например, известное видео "Mac or PC" шуточно показывает преимущество Макинтош над Виндовз, и большим антивирусным иммунитетом Мак-ОС против системы от Майкрософт[1].

Помимо ОС для настольных компьютеров и ноутбуков, также существуют платформы и для мобильных устройств, такие как Windows Mobile, Symbian, iOS, BlackBerry, Android и др. Пользователи устройств на данных ОС также подвержены риску заражения вредоносным программным обеспечением, поэтому некоторые разработчики антивирусных программ выпускают продукты и для таких устройств.

Классификация антивирусных продуктов

Классифицировать антивирусные продукты можно сразу по нескольким признакам, таким как: используемые технологии антивирусной защиты, функционал продуктов, целевые платформы.

По используемым технологиям антивирусной защиты:

Классические антивирусные продукты (продукты, применяющие только сигнатурный метод детектирования)

Продукты проактивной антивирусной защиты (продукты, применяющие только проактивные технологии антивирусной защиты);

Комбинированные продукты (продукты, применяющие как классические, сигнатурные методы защиты, так и проактивные)

По функционалу продуктов:

Антивирусные продукты (продукты, обеспечивающие только антивирусную защиту)

Комбинированные продукты (продукты, обеспечивающие не только защиту от вредоносных программ, но и фильтрацию спама, шифрование и резервное копирование данных и другие функции)

По целевым платформам:
Антивирусные продукты для ОС семейства Windows
Антивирусные продукты для ОС семейства *UNIX (к данному семейству относятся ОС BSD, Linux, Mac OS X и др.)
Антивирусные продукты для мобильных платформ (Windows Mobile, Symbian, iOS, BlackBerry, Android и др.)

Антивирусные продукты для корпоративных пользователей можно также классифицировать по объектам защиты:
Антивирусные продукты для защиты рабочих станций
Антивирусные продукты для защиты файловых и терминальных серверов
Антивирусные продукты для защиты почтовых и Интернет-шлюзов
Антивирусные продукты для защиты серверов виртуализации
и др.
Работа антивируса

Говоря о системах Майкрософт, обычно антивирус действует по схеме: - поиск в базе данных антивирусного ПО "сигнатур" вирусов - если найден инфицированный код в памяти - или оперативной и/или постоянной - запускается процесс карантина и процесс блокируется - зарегистрированная программа обычно удаляет вирус, незарегистрированная просит регистрации, и оставляет систему уязвимой

Базы антивирусов

Для использования антивирусов, необходимы постоянные обновления так называемых баз антивирусов. Они представляют собой информацию о вирусах - как их найти и обезвредить. Поскольку вирусы пушут часто, то необходим постоянный мониторинг активности вирусов в сети. Для этого существуют специальные сети которые собирают соответствующую информацию. После сбора этой информации производится анализ вредоносности вируса, анализируется его код, поведение, и после этого устанавливаются способы борьбы с ним. Чаще всего вирусы запускаются вместе с операционной системой, и появляются записи в системном реестре. В таком случае можно просто удалить строки запуска вируса из реестра, и на этом обычно процесс может закончиться в простом случае. Более сложные вирусы используют возможность заражения файлов. Например, известны случаи, как некие даже антивирусные программы, будучи зараженными - сами становились причиной заражения других чистых программ и файлов. Поэтому более современные антивирусы имеют возможность защиты своих файлов от изменения, и проверяют их на целостность по специальному алгоритму. Таким образом вирусы усложнились, как и усложнились способы борьбы с ними. Сейчас можно видеть в интернете вирусы которые занимают уже не десятки килобайт, а сотни, а порой могут быть и размером в парочку мегабайт. Обычно такие вирусы пишут в языках программирования более высокого уровня, поэтому их легче остановить. Но по прежнему существует угроза от вирусов написанных на низкоуровневых машинных кодах наподобие ассемблера. Сложные вирусы заражают операционную систему, после чего она становится уязвимой и нерабочей. Особенную опасность представляют пиратское ПО, поскольку оно в себе предполагает наличие вредоносного кода (в том числе и вред для интеллектуальной собственности). Таким образом, пользуясь пиратским антивирусом, человек рискует поте-

рять заработанные деньги и поставить бизнес под угрозу. Поэтому очень важно иметь лицензионный (можно и бесплатный) антивирус.

Вместе с тем решение этого вопроса достигается путём сочетания организационных мер и программно-технических решений. Данный подход не требует больших технических и немедленных финансовых затрат, и может быть применён для комплексной антивирусной защиты локальной сети любого предприятия.

В основу построения такой системы антивирусной защиты могут быть положены следующие принципы:
принцип реализации единой технической политики при обосновании выбора антивирусных продуктов для различных сегментов локальной сети;
принцип полноты охвата системой антивирусной защиты всей локальной сети организации;
принцип непрерывности контроля локальной сети предприятия, для своевременного обнаружения компьютерной инфекции;
принцип централизованного управления антивирусной защитой;

Принцип реализации единой технической политики предусматривает использование во всех сегментах локальной сети только антивирусного ПО, рекомендуемого подразделением антивирусной защиты предприятия. Эта политика носит долгосрочный характер, утверждается руководством предприятия и является основой для целевого и долговременного планирования затрат на приобретение антивирусных программных продуктов и их дальнейшее обновление.

Принцип полноты охвата системой антивирусной защиты локальной сети предусматривает постепенной внедрение в сеть программных средств антивирусной защиты до полного насыщения в сочетании с организационно-режимными мерами защиты информации.

Принцип непрерывности контроля за антивирусным состоянием локальной сети подразумевает такую организацию ее защиты, при которой обеспечивается постоянная возможность отслеживания состояния сети для выявления вирусов.

Принцип централизованного управления антивирусной защитой предусматривает управление системой из одной органа с использованием технических и программных средств. Именно этот орган организует централизованный контроль в сети, получает данные контроля или доклады пользователей со своих рабочих мест об обнаружении вирусов и обеспечивает внедрение принятых решений по управлению системой антивирусной защиты.

С учётом этих принципов в комплексной системе информационной безопасности создаётся подразделение антивирусной защиты, которая должна решать следующие задачи:
приобретение, установка и своевременная замена антивирусных пакетов на серверах и рабочих станциях пользователей;
контроль правильности применения антивирусного ПО пользователями;
обнаружение вирусов в локальной сети, их оперативное лечение, удаление зараженных объектов, локализация зараженных участков сети;
своевременное оповещение пользователей об обнаруженных или возможных вирусах, их признаках и характеристиках.

Для решения этих задач в комплексной системе информационной безопасности кроме администраторов информационной безопасности создаются администраторы антивирусной защиты. Если ЛВС небольшая или достаточно хорошо оснащена антивирусным ПО, то назначение специального администратора антивирусной защиты чаще всего нецелесообразно, так как его функции может выполнять администратор безопасности сети.

Для организации функционирования антивирусной защиты необходима разработка внутренних организационно-распорядительных документов. Кроме того, должны быть определены порядки передачи сообщений о вирусах от пользователей и оповещений администраторов о фактах и возможностях вирусных заражений локальной сети.

Эффективность создаваемой подсистемы антивирусной защиты зависит также от выполнения следующих дополнительных условий:
подключение ПК пользователей в корпоративную сеть должно производиться только по заявке с отметкой администратора антивирусной защиты об установке лицензионного антивирусного ПО (заявка заносится в базу данных с фиксацией сроков действия лицензии);
передачу ПК от одного пользователя другому необходимо производить с переоформлением подключения к сети;
обнаруженные вирусы целесообразно исследовать на стенде подразделения защиты информации с целью выработки рекомендаций по их корректному обезвреживанию;
в удаленных структурных подразделениях следует назначить внештатных сотрудников, ответственных за антивирусную защиту.

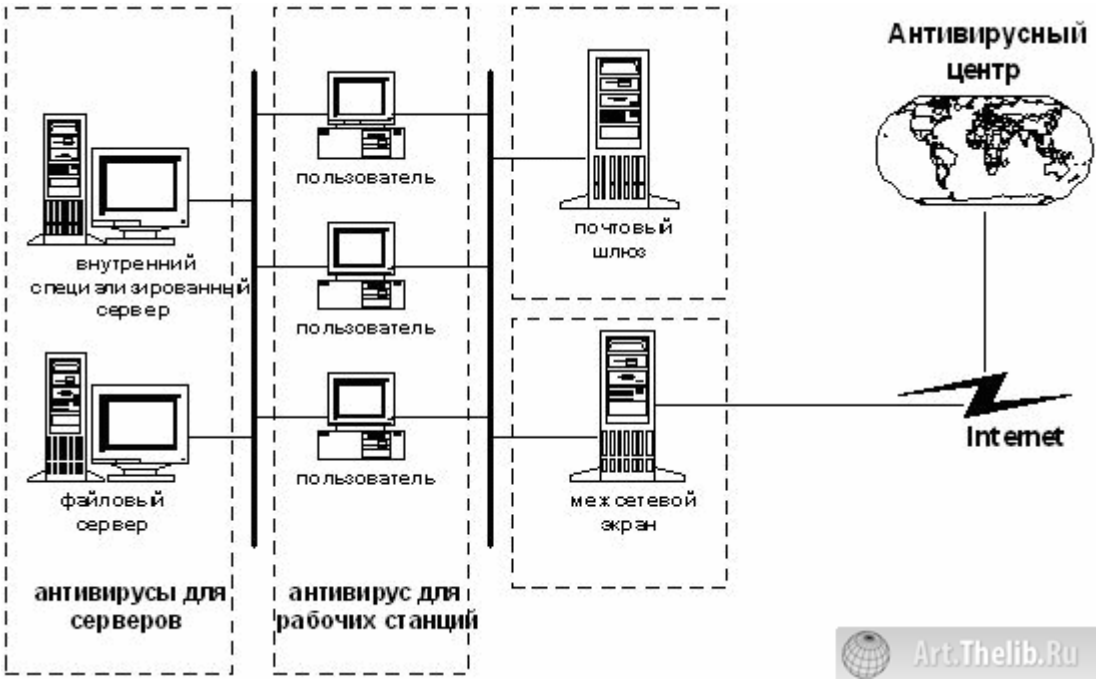
Практическая реализация антивирусной защиты информации на серверах и ПК корпоративной сети осуществляется с использованием ряда программно-технических методов, являющихся стандартными, но имеющих свою специфику, определяемую особенностями корпоративной сети. К ним относятся:
использование антивирусных пакетов;
архивирование информации;
резервирование информации;
ведение базы данных о вирусах и их характеристиках;

Рассмотрим эти методы более подробно.

Главным методом антивирусной защиты является установка антивирусных пакетов. Выбор антивирусного ПО является одной из важнейших задач антивирусной защиты, от правильности решения которой в дальнейшем будут зависеть антивирусная безопасность системы, а также затраты на ее поддержание. Используемые антивирусные средства должны удовлетворять следующим общим требованиям:
система должны быть совместима с операционными системами серверов и ПК;
система антивирусной защиты не должна нарушать логику работы остальных используемых приложений;
наличие полного набора антивирусных функций, необходимых для обеспечения антивирусного контроля и обезвреживания всех известных вирусов;
частота обновления антивирусного ПО и гарантии поставщиков (разработчиков) в отношении его своевременности.

В отличие от других подсистем информационной безопасности в рассматриваемой области отсутствуют четко сформулированные показатели защищенности и соответствующие критерии сравнения различных антивирусных средств. Как правило антивирусные комплексы сравниваются по следующим показателям: обнаружение, лечение, блокирование, восстановление, регистрация, обеспечение целостности, обновление базы данных компьютерных вирусов, защита антивирусных средств от доступа паролем, средства управления, гарантии проектирования, документация.

При комплексной защите локальной сети необходимо уделить внимание всем возможным точкам проникновения вирусов в сеть извне.



На рисунке 1 приведена общая структура антивирусной защиты локальной сети. На первом уровне защищают подключение в Интернет или сеть поставщика услуг связи - это межсетевой экран и почтовые шлюзы, поскольку по статистике именно оттуда попадает около 80% вирусов. Необходимо отметить, что таким образом будет обнаружено не более 30% вирусов, так как оставшиеся 70% будут обнаружены только в процессе выполнения.

Применение антивирусов для межсетевых экранов на сегодняшний день сводится к осуществлению фильтрации доступа в Интернет при одновременной проверке на вирусы проходящего трафика. Осуществляемая такими продуктами антивирусная проверка сильно замедляет работу и имеет крайне не высокий уровень обнаружения, по этому в отсутствии необходимости фильтрации посещаемых пользователями веб-узлов применение таких продуктов является не целесообразным.

Антивирусной защите подлежат все компоненты информационной системы, участвующие в транспортировке информации и/или её хранении:
файл-серверы;
рабочие станции;
рабочие станции мобильных пользователей;
сервера резервного копирования;
почтовые сервера.

Как правило, использование одного (базового) антивирусного пакета для защиты локальной сети представляется наиболее целесообразным. Однако анализ рынка антивирусных средств показывает, что в случае, когда мы имеем дело с большой корпоративной сетью, это не всегда возможно вследствие разнородности применяемых в сегментах сети операционных платформ.

Следующим после выбора пакетов шагом является их тестирование администратором безопасности на специальном стенде подразделения защиты информации. Эта процедура позволяет выявить ошибки в антивирусном ПО, оценить его совместимость с системным и прикладным ПО, используемым на ПК и серверах сети. Опыт показывает, что такое тестирование оказывается далеко не лишним, поскольку разработчик не способен в полном объеме исследовать процесс функционирования своих антивирусных средств в условиях реальных сетей. Результаты тестирования направляются разработчику пакета, что позволяет тому провести необходимые доработки до начала массовой установки последнего.

Современные антивирусные пакеты содержат в себе следующие основные программные компоненты:
монитор (резидентно размещается в оперативной памяти компьютера и автоматически проверяет объекты перед их запуском или открытием; при обнаружении вируса программа в зависимости от настроек может: удалить зараженный объект, вылечить его, запретить к нему доступ);
сканер (осуществляет проверку объектов на наличие вирусов по запросу пользователей);
сетевой центр управления (позволяет организовать управление АВЗ корпоративной сети: управлять компонентами пакета, задавать расписания запуска сканера, автоматического обновления антивирусных баз и т.д.);
дополнительные модули, обеспечивающие проверку электронной почты и Web-страниц в момент получения информации.

Установку антивирусных пакетов и их настройку выполняют специалисты подразделения, осуществляющего техническое обслуживание сети. Программы "монитор" и "сканер" устанавливаются как на серверах, так и на ПК, причем первый настраивается на постоянное включение.

При обнаружении вирусов пользователям не рекомендуется заниматься "самолечением", так как это может привести к потере информации. В таких случаях им следует по "горячей линии" обращаться к администраторам антивирусной защиты, которые принимают меры по обезвреживанию вирусов и предотвращению дальнейшего заражения.

Следующими по важности методами антивирусной защиты являются архивирование и резервное копирование информации, позволяющие исключить потерю информации в случае вирусного заражения. Архивирование заключается в периодическом копировании системных областей машинных носителей информации на внешние устройства. На серверах с наиболее важной информацией архивирование необходимо проводить с минимальной периодичностью. Резервное копирование информации проводится ежедневно в целях защиты ее от искажения и разрушения.

Раздел 2

1. Протокол ICMP. Модель, основные команды, безопасность, производительность.

Данный протокол на ряду с IP и ARP относят к межсетевому уровню. Протокол используется для рассылки информационных и управляющих сообщений. При этом используются следующие виды сообщений:

Flow control - если принимающая машина (шлюз или реальный получатель информации) не успевает перерабатывать информацию, то данное сообщение приостанавливает отправку пакетов по сети.

Detecting unreachable destination - если пакет не может достичь места назначения, то шлюз, который не может доставить пакет, сообщает об этом отправителю пакета. Информировать о невозможности доставки сообщения может и машина, чей IP-адрес указан в пакете. Только в этом случае речь будет идти о портах TCP и UDP, о чем будет сказано чуть позже.

Redirect routing - это сообщение посылается в том случае, если шлюз не может доставить пакет, но у него есть на этот счет некоторые соображения, а именно адрес другого шлюза.

Checking remote host - в этом случае используется так называемое ICMP Echo Message. Если необходимо проверить наличие стека TCP/IP на удаленной машине, то на нее посылается сообщение этого типа. Как только система получит это сообщение, она немедленно подтвердит его получение.

Последняя возможность широко используется в Internet. На ее основе работает команда ping.

Другое использование ICMP - это получение сообщения о "кончине" пакета на шлюзе. При этом используется время жизни пакета, которое определяет число шлюзов, через которые пакет может пройти. Программа, которая использует этот прием, называется traceroute. К более подробному обсуждению ее возможностей мы вернемся в разделе 2.5. Здесь же только укажем, что она использует сообщение TIME EXECEED протокола ICMP.

При посылке пакета через Internet traceroute устанавливает значение TTL (Time To Live) последовательно от 1 до 30 (значение по умолчанию). TTL определяет число шлюзов, через которые может пройти IP-пакет. Если это число превышено, то шлюз, на котором происходит обнуление TTL, высылает ICMP-пакет. Traceroute сначала устанавливает значение TTL равное единице - отвечает ближайший шлюз, затем значение TTL равно 2 - отвечает следующий шлюз и т. д. Если пакет достиг получателя, то в этом случае возвращается сообщение другого типа - Detecting unreachable destination, т.к. IP-пакет передается на транспортный уровень, а на нем нет обслуживания запросов traceroute.

Протокол ICMP является неотъемлемой частью IP-модуля. Он обеспечивает обратную связь в виде диагностических сообщений, посылаемых отправителю при невозможности доставки его дейтаграммы и в других случаях. ICMP стандартизован в RFC-792, дополнения — в RCF-950,1256.

ICMP-сообщения не порождаются при невозможности доставки:

- дейтаграмм, содержащих ICMP-сообщения;
- не первых фрагментов дейтаграмм;
- дейтаграмм, направленных по групповому адресу (широковещание, мультикастинг);
- дейтаграмм, адрес отправителя которых нулевой или групповой.

Все ICMP-сообщения имеют IP-заголовок, значение поля “Protocol” равно 1. Данные дейтаграммы с ICMP-сообщением не передаются вверх по стеку протоколов для обработки, а обрабатываются IP-модулем.

После IP-заголовка следует 32-битное слово с полями “Тип”, “Код” и “Контрольная сумма”. Поля типа и кода определяют содержание ICMP-сообщения. Формат остальной части дейтаграммы зависит от вида сообщения. Контрольная сумма считается так же, как и в IP-заголовке, но в этом случае суммируется содержимое ICMP-сообщения, включая поля “Тип” и “Код”.

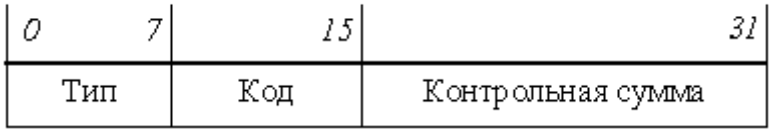


Рисунок 11 Дейтаграмма ICMP-сообщения

Тип	Код	Сообщение
0	0	Echo Reply (эхо-ответ)
3		Destination Unreachable (адресат недостижим по различным причинам):
	0	Net Unreachable (сеть недоступна)
	1	Host Unreachable (хост недоступен)
	2	Protocol Unreachable (протокол недоступен)
	3	Port Unreachable (порт недоступен)
	4	DF=1 (необходима фрагментация, но она запрещена)
	5	Source Route failed (невозможно выполнить опцию Source Route)
4	0	Source Quench (замедление источника)
5		Redirect (выбрать другой маршрутизатор для посылки дейтаграмм)
	0	в данную сеть
	1	на данный хост
	2	в данную сеть с данным TOS
	3	на данный хост с данным TOS
8	0	Echo Request (эхо-запрос)
9	0	Router Advertisement (объявление маршрутизатора)
10	0	Router Solicitation (запрос объявления маршрутизатора)
11		Time Exceeded (время жизни дейтаграммы истекло)
	0	при передаче
	1	при сборке
12		Parameter problem (ошибка в параметрах)
	0	Ошибка в IP-заголовке
	1	Отсутствует необходимая опция
13	0	Timestamp (запрос временной метки)
14	0	Timestamp Reply (ответ на запрос временной метки)
17	0	Address Mask Request (запрос сетевой маски)
18	0	Address Mask Reply (ответ на запрос сетевой маски)

Ниже рассмотрены форматы ICMP-сообщений и даны комментарии к некоторым сообщениям.

Типы 3, 4, 11, 12

В сообщении типа 12 в поле “xxxxxxxxxx” (1 октет) заносится номер октета заголовка, в котором обнаружена ошибка; в сообщениях типов 3, 4, 11 не используется. Все неиспользуемые поля заполняются нулями.

Сообщения типа 4 (“Замедление источника”) генерируются в случае переполнения (или опасности переполнения) буферов обработки дейтаграмм адресата или промежуточного узла на маршруте. При получении такого сообщения отправитель должен уменьшить скорость или приостановить отправку дейтаграмм до тех пор, пока он не перестанет получать сообщения этого типа.

IP-заголовок и начальные слова оригинальной дейтаграммы приводятся для опознания ее отправителем и, возможно, анализа причины сбоя.

Тип 5

Сообщения типа 5 направляются маршрутизатором отправителю дейтаграммы в случае, когда маршрутизатор считает, что дейтаграммы в данное место назначения следует направлять через другой маршрутизатор. Адрес нового маршрутизатора приведен во втором слове сообщения.

Понятие “место назначения” конкретизируется значением поля “Код” (см. табл. 2.5.1). Информация о том, куда была направлена дейтаграмма, породившая ICMP-сообщения, извлекается из ее заголовка, присоединенного к сообщению. Отсутствие передачи сетевой маски ограничивает область применения сообщений типа 5.

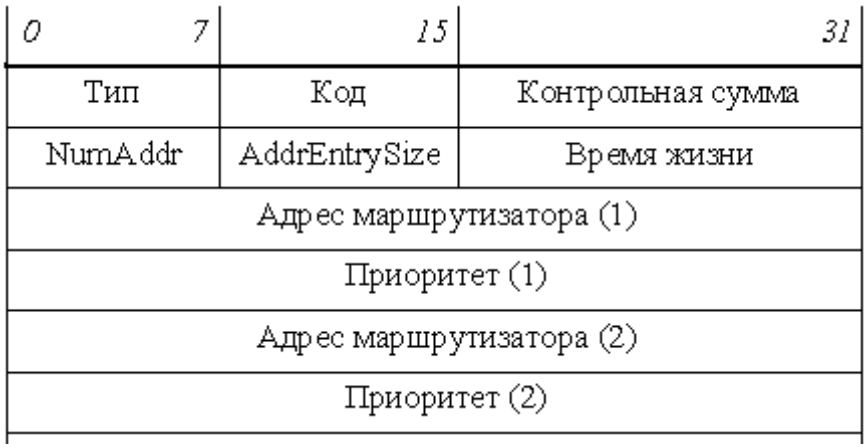
Типы 0,8

Сообщения типов 0 и 8 используются для тестирования связи по протоколу IP между двумя узлами сети. Тестирующий узел генерирует сообщения типа 8 (“Эхо-запрос”), при этом “Идентификатор” определяет данный сеанс тестирования (номер последовательности отправляемых сообщений), поле “Номер по порядку” содержит номер данного сообщения внутри последовательности. В поле данных содержатся произвольные данные, размер этого поля определяется общей длиной дейтаграммы, указанной в поле “Total length” IP-заголовка.

IP-модуль, получивший эхо-запрос, отправляет эхо-ответ. Для этого он меняет местами адреса отправителя и получателя, изменяет тип ICMP-сообщения на 0 и пересчитывает контрольную сумму.

Тестирующий узел по самому факту получения эхо-ответов, времени оборота дейтаграмм, проценту потерь и последовательности прибытия ответов может сделать выводы о наличии и качестве связи с тестируемым узлом. На основе посылки и приема эхо-сообщений работает программа ping.

Тип 9



Сообщения типа 9 (объявление маршрутизатора) периодически рассылаются маршрутизаторами хостам сети для того, чтобы хосты могли автоматически сконфигурировать свои маршрутные таблицы. Обычно такие сообщения рассылаются по мультикастинговому адресу 224.0.0.1 (“всем хостам”) или по широковещательному адресу.

Сообщение содержит адреса одного или нескольких маршрутизаторов, снабженных значениями приоритета для каждого маршрутизатора. Приоритет является числом со знаком, записанным в дополнительном коде; чем больше число, тем выше приоритет.

Поле “NumAddr” содержит количество адресов маршрутизаторов в данном сообщении; значение поля “AddrEntrySize” равно двум (размер поля, отведенного на информацию об одном маршрутизаторе, в 32-битных словах). “Время жизни” определяет срок годности информации, содержащейся в данном сообщении, в секундах.

Тип 10

Сообщения типа 10 (запрос объявления маршрутизатора) состоит из двух 32-битных слов, первое из которых содержит поля “Тип”, “Код” и “Контрольная сумма”, а второе зарезервировано (заполняется нулями).

Типы 17 и 18

Сообщения типов 17 и 18 (запрос и ответ на запрос значения маски сети) используются в случае, когда хост желает узнать маску сети, в которой он находится. Для этого в адрес маршрутизатора (или широковещательно, если адрес маршрутизатора неизвестен) отправляется запрос. Маршрутизатор отправляет в ответ сообщение с записанным в нем значением маски той сети, из которой пришел запрос. В том случае, когда отправитель запроса еще не знает своего IP-адреса, ответ отправляется широковещательно.

Поля “Идентификатор” и “Номер по порядку” могут использоваться для контроля соответствий запросов и ответов, но в большинстве случаев игнорируются.

2. Сетевые архитектуры: понятие, сравнительные характеристики.

Сетевая архитектура (network architecture) - это комбинация стандартов, топологий и протоколов, необходимых для создания работоспособной сети.

2.1 Ethernet

В сетях Ethernet используется метод доступа к среде передачи данных, называемый методом коллективного доступа с опознаванием несущей и обнаружением коллизий (carrier-sense-multiply-access with collision detection, CSMA/ CD).

Этот метод применяется исключительно в сетях с логической общей шиной (к которым относятся и радиосети, породившие этот метод). Все компьютеры такой сети имеют непосредственный доступ к общей среде, поэтому она может быть использована для передачи данных между любыми двумя узлами сети. Одновременно все компьютеры сети имеют возможность немедленно (с учетом задержки распространения сигнала по физической среде) получить данные, которые любой из компьютеров начал передавать в общую среду.

Простота схемы подключения — один из факторов, определивших успех стандарта Ethernet. Говорят, что среда, к которой подключены все станции, работает в режиме коллективного доступа (Multiply Access, MA).

На MAC-уровне для идентификации сетевых интерфейсов узлов сети используются регламентированные стандартом IEEE 802.3 уникальные 6-байтовые адреса, называемые MAC-адресами. Каждый сетевой адаптер имеет, по крайней мере, один MAC-адрес.

Чтобы получить возможность передавать кадр, интерфейс-отправитель должен убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая также называется несущей частотой (carrier-sense, CS). Признаком незанятости среды является отсутствие на ней несущей частоты, которая при манчестерском способе кодирования равна 5-10 МГц, в зависимости от последовательности единиц и нулей, передаваемых в данный момент. Узел 1 обнаружил, что среда свободна, и начал передавать свой кадр. В классической сети Ethernet на коаксиальном кабеле сигналы передатчика узла 1 распространяются в обе стороны, так что все узлы сети их получают. Кадр данных всегда сопровождается преамбулой (preamble), которая состоит из 7 байтов, состоящих из значений 10101010, и 8-го байта, равного 10101011. Последний байт носит название ограничителя начала кадра - Start of Frame. Преамбула нужна для вхождения приемника в побитовый и побайтовый синхронизм с передатчиком.

Все станции, подключенные к кабелю, могут распознать факт передачи кадра, и та станция, которая узнает собственный адрес в заголовках кадра, записывает его содержимое в свой внутренний буфер, обрабатывает полученные данные, передает их вверх по своему стеку, а затем посылает по кабелю кадр-ответ. Адрес станции-источника содержится в исходном кадре, поэтому станция-получатель знает, кому нужно послать ответ.

Узел 2 во время передачи кадра узлом 1 также пытался начать передачу своего кадра, однако обнаружил, что среда занята - на ней присутствует несущая частота, - поэтому узел 2 вынужден ждать, пока узел 1 не прекратит передачу кадра.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу (Inter Packet Gap) в 9,6 мкс. Эта пауза, называемая также межкадровым интервалом, нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одной станцией. После окончания технологической паузы узлы имеют право начать передачу своего кадра, так как среда свободна. Из-за задержек распространения сигнала по кабелю не все узлы строго одновременно фиксируют факт окончания передачи кадра узлом 1.

При описанном подходе возможна ситуация, когда две станции одновременно пытаются передать кадр данных по общей среде. Механизм прослушивания среды и пауза между кадрами не гарантируют исключения такой ситуации, когда две или более станции одновременно решают, что среда свободна, и начинают передавать свои кадры. Говорят, что при этом происходит коллизия (collision), так как содержимое обоих кадров сталкивается на общем кабеле и происходит искажение информации - методы кодирования, используемые в Ethernet, не позволяют выделять сигналы каждой станции из общего сигнала.

2.2 Token Ring

Сети Token Ring (стандарт 802.5), так же как и сети Ethernet, характеризует разделяемая среда передачи данных, которая в данном случае состоит из отрезков кабеля, соединяющих все станции сети в кольцо. Кольцо рассматривается как общий разделяемый ресурс, и для доступа к нему требуется не случайный алгоритм, как в сетях Ethernet, а детерминированный, основанный на передаче станциям права на использование кольца в определенном порядке. Это право передается с помощью кадра специального формата, называемого маркером, или токеном, (token).

В сети Token Ring любая станция всегда непосредственно получает данные только от одной станции - той, которая является предыдущей в кольце. Такая станция называется ближайшим активным соседом, расположенным выше по потоку (данных) - Nearest Active Upstream Neighbor, NAUN. Передачу же данных станция всегда осуществляет своему ближайшему соседу вниз по потоку данных.

Получив маркер, станция анализирует его и при отсутствии у нее данных для передачи обеспечивает его продвижение к следующей станции. Станция, которая имеет данные для передачи, при получении маркера изымает его из кольца, что дает ей право доступа к физической среде для передачи своих данных. Затем эта станция выдает в кольцо кадр данных установленного формата последовательно по битам. Переданные данные проходят по кольцу всегда в одном направлении от одной станции к другой. Кадр снабжен адресом назначения и адресом источника.

Все станции кольца ретранслируют кадр побитно, как повторители. Если кадр проходит через станцию назначения, то, распознав свой адрес, эта станция копирует кадр в свой внутренний буфер и вставляет в кадр признак подтверждения приема. Станция, выдавшая кадр данных в кольцо, при обратном его получении с подтверждением приема изымает этот кадр из кольца и передает в сеть новый маркер, давая другим станциям сети возможность передавать данные. Такой алгоритм доступа применяется в сетях Token Ring со скоростью работы 4 Мбит/с, описанных в стандарте 802.5. В сетях Token Ring 16 Мбит/с используется также несколько другой алгоритм доступа к кольцу, называемый алгоритмом раннего освобождения маркера (Early Token Release). В соответствии с ним станция передает маркер доступа следующей станции сразу же после окончания передачи последнего бита кадра, не дожидаясь возвращения по кольцу этого кадра с битом подтверждения приема. В этом случае пропускная способность кольца используется более эффективно, так как по кольцу одновременно продвигаются кадры нескольких станций. Тем не менее свои кадры в каждый

момент времени может генерировать только одна станция - та, которая в данный момент владеет маркером доступа. Остальные станции в это время только повторяют чужие кадры, так что принцип разделения кольца во времени сохраняется, ускоряется только процедура передачи владения кольцом.

Для различных видов сообщений, передаваемым кадрам, могут назначаться различные приоритеты: от 0 (низший) до 7 (высший). Решение о приоритете конкретного кадра принимает передающая станция (протокол Token Ring получает этот параметр через межуровневые интерфейсы от протоколов верхнего уровня, например прикладного). Маркер также всегда имеет некоторый уровень текущего приоритета. Станция имеет право захватить переданный ей маркер только в том случае, если приоритет кадра, который она хочет передать, выше (или равен) приоритета маркера. В противном случае станция обязана передать маркер следующей по кольцу станции.

Сети Token Ring работают с двумя битовыми скоростями - 4 и 16 Мбит/с. Смешение станций, работающих на различных скоростях, в одном кольце не допускается. Сети Token Ring, работающие со скоростью 16 Мбит/с, имеют некоторые усовершенствования в алгоритме доступа по сравнению со стандартом 4 Мбит/с. Технология Token Ring является более сложной технологией, чем Ethernet. Она обладает свойствами отказоустойчивости. В сети Token Ring определены процедуры контроля работы сети, которые используют обратную связь кольцеобразной структуры - посланный кадр всегда возвращается в станцию- отправитель. В некоторых случаях обнаруженные ошибки в работе сети устраняются автоматически, например может быть восстановлен потерянный маркер. В других случаях ошибки только фиксируются, а их устранение выполняется вручную обслуживающим персоналом.

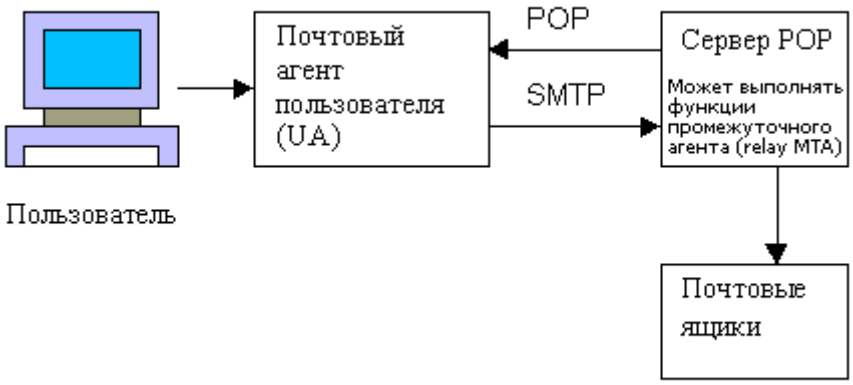
Для контроля сети одна из станций выполняет роль так называемого активного монитора. Активный монитор выбирается во время инициализации кольца как станция с максимальным значением MAC-адреса. Если активный монитор выходит из строя, процедура инициализации кольца повторяется и выбирается новый активный монитор. Чтобы сеть могла обнаружить отказ активного монитора, последний в работоспособном состоянии каждые три секунды генерирует специальный кадр своего присутствия. Если этот кадр не появляется в сети более 7 с, то остальные станции сети начинают процедуру выборов нового активного монитора.

Сравнительные характеристики ETHERNET и Token Ring

Характеристика	FDDI	Ethernet	Token Ring
Битовая скорость	100 Мбит/с	10 Мбит/с	16 Мбит/с
Топология	Двойное кольцо деревьев	Шина/звезда	Звезда/кольцо
Метод доступа	Доля от времени оборота маркера	CSMA/CD	Приоритетная система резервирования
Среда передачи данных	Оптоволокно, неэкранированная витая пара категории 5	Толстый коаксиал, тонкий коаксиал, витая пара категории 3, оптоволокно	Экранированная и неэкранированная витая пара, оптоволокно
Максимальная длина сети (без мостов)	200 км (100 км на кольцо)	2500 м	4000 м
Максимальное расстояние между узлами	2 км (не больше 11 дБ потерь между узлами)	2500 м	100 м
Максимальное количество узлов	500 (1000 соединений)	1024	260 для экранированной витой пары, 72 для неэкранированной витой пары
Тактирование и восстановление после отказов	Распределенная реализация тактирования и восстановления после отказов	Не определены	Активный монитор

3. Протокол POP. Модель, основные команды, безопасность, производительность.

Post Office Protocol (POP) - протокол доставки почты пользователю из почтового ящика почтового сервера POP. Многие концепции, принципы и понятия протокола POP выглядят и функционируют подобно SMTP. На рисунке изображена модель клиент-сервер по протоколу POP. Сервер POP находится между агентом пользователя и почтовыми ящиками.



Конфигурация модели клиент-сервер по протоколу POP

В настоящее время существуют две версии протокола POP - POP2 и POP3, обладающими примерно одинаковыми возможностями, однако несовместимыми друг с другом. Дело в том, что у POP2 и POP3 разные номера портов протокола. Между ними отсутствует связь, аналогичная связи между SMTP и ESMTP. Протокол POP3 не является расширением или модификацией POP2 - это совершенно другой протокол. POP2 определен в документе RFC 937 (Post Office Protocol-Version 2, Butler, et al, 1985), а POP3 - в RFC 1225 (Post Office Protocol-Version 3, Rose, 1991). Далее кратко рассмотрим POP вообще и более подробно - POP3. POP3 разработан с учетом специфики доставки почты на персональные компьютеры и имеет соответствующие операции для этого.

Назначение протокола POP3

Ранее почтовые сообщения большинства сетей доставлялись непосредственно от одного компьютера к другому. И если пользователь часто менял рабочие компьютеры или один компьютер принадлежал нескольким пользователям, существовали определенные проблемы. В наши дни общепринята доставка сообщения не на компьютеры пользователя, а в специальные почтовые ящики почтового сервера организации, который круглосуточно работает (включен).

Описание протокола POP3

Конструкция протокола POP3 обеспечивает возможность пользователю обратиться к своему почтовому серверу и изъять накопившуюся для него почту. Пользователь может получить доступ к POP-серверу из любой точки доступа к Интернет. При этом он должен запустить специальный почтовый агент (UA), работающий по протоколу POP3, и настроить его для работы со своим почтовым сервером. Итак, во главе модели POP находится отдельный персональный компьютер, работающий исключительно в качестве клиента почтовой системы (сервера). Разработчики протокола POP3 называет такую ситуацию "раздельные агенты" (split UA). Концепция раздельных агентов кратко обсуждается в спецификации POP3.

В протоколе POP3 оговорены три стадии процесса получения почты: авторизация, транзакция и обновление. После того как сервер и клиент POP3 установили соединение, начинается стадия авторизации. На стадии авторизации клиент идентифицирует себя для сервера. Если авторизация прошла успешно, сервер открывает почтовый ящик клиента и начинается стадия транзакции. В ней клиент либо запрашивает у сервера информацию (например, список почтовых сообщений), либо просит его совершить определенное действие (например, выдать почтовое сообщение). Наконец, на стадии обновления сеанс связи заканчивается. В табл.7 перечислены команды протокола POP3, обязательные для работающей в Интернет реализации минимальной конфигурации.

Таблица 5. Команды протокола POP версии 3 (для минимальной конфигурации)

Команда	Описание
USER	Идентифицирует пользователя с указанным именем
PASS	Указывает пароль для пары клиент-сервер
QUIT	Закрывает TCP-соединение
STAT	Сервер возвращает количество сообщений в почтовом ящике плюс размер почтового ящика
LIST	Сервер возвращает идентификаторы сообщений вместе с размерами сообщений (параметром команды может быть идентификатор сообщения)
RETR	Извлекает сообщение из почтового ящика (требуется указывать аргумент-идентификатор сообщения)
DELE	Отмечает сообщение для удаления (требуется указывать аргумент - идентифика-

	тор сообщения)
NOOP	Сервер возвращает положительный ответ, но не совершает никаких действий
LAST	Сервер возвращает наибольший номер сообщения из тех, к которым ранее уже обращались
RSET	Отменяет удаление сообщения, отмеченного ранее командой DELE

В протоколе POP3 определено несколько команд, но на них дается только два ответа: +OK (позитивный, аналогичен сообщению-подтверждению ACK) и -ERR (негативный, аналогичен сообщению "не подтверждено" NAK). Оба ответа подтверждают, что обращение к серверу произошло и что он вообще отвечает на команды. Как правило, за каждым ответом следует его содержательное словесное описание. В RFC 1225 есть образцы нескольких типичных сеансов POP3. Сейчас мы рассмотрим несколько из них, что даст возможность уловить последовательность команд в обмене между сервером и клиентом.

Авторизация пользователя
После того как программа установила TCP-соединение с портом протокола POP3 (официальный номер 110), необходимо послать команду USER с именем пользователя в качестве параметра. Если ответ сервера будет +OK, нужно послать команду PASS с паролем этого пользователя:

CLIENT: USER kcope
SERVER: +OK
CLIENT: PASS secret
SERVER: +OK kcope's maildrop has 2 messages (320 octets)
(В почтовом ящике kcope есть 2 сообщения (320 байтов) ...)

Транзакции POP3
После того как стадия авторизации окончена, обмен переходит на стадию транзакции. В следующих примерах демонстрируется возможный обмен сообщениями на этой стадии.
Команда STAT возвращает количество сообщений и количество байтов в сообщениях:
CLIENT: STAT
SERVER: +OK 2 320
Команда LIST (без параметра) возвращает список сообщений в почтовом ящике и их размеры:
CLIENT: LIST
SERVER: +OK 2 messages (320 octets)
SERVER: 1 120
SERVER: 2 200
SERVER: . ...
Команда LIST с параметром возвращает информацию о заданном сообщении:
CLIENT: LIST 2
SERVER: +OK 2 200 ...
CLIENT: LIST 3
SERVER: -ERR no such message,
only 2 messages in maildrop
Команда TOP возвращает заголовок, пустую строку и первые десять строк тела сообщения:
CLIENT: TOP 10
SERVER: +OK
SERVER: <the POP3 server sends the headers
of the message,a blank line, and the first 10 lines
of the message body>

(сервер POP высылает заголовки сообщений,
пустую строку и первые десять строк тела сообщения)
SERVER: . ...
CLIENT: TOP 100
SERVER: -ERR no such message
Команда NOOP не возвращает никакой полезной информации, за исключением позитивного ответа сервера. Однако позитивный ответ означает, что сервер находится в соединении с клиентом и ждет запросов:
CLIENT: NOOP
SERVER: +OK
Следующие примеры показывают, как сервер POP3 выполняет действия. Например, команда RETR извлекает сообщение с указанным номером и помещает его в буфер местного UA:
CLIENT: RETR 1
SERVER: +OK 120 octets
SERVER: <the POPS server sends
the entire message here>
(POP3-сервер высылает сообщение целиком)
SERVER: . . . . . .
Команда DELE отмечает сообщение, которое нужно удалить:
CLIENT: DELE 1
SERVER: +OK message 1 deleted ...
(сообщение 1 удалено)
CLIENT: DELE 2
SERVER: -ERR message 2 already deleted
сообщение 2 уже удалено)
Команда RSET снимает метки удаления со всех отмеченных ранее сообщений:
CLIENT: RSET
SERVER: +OK maildrop has 2 messages (320 octets)
(в почтовом ящике 2 сообщения (320 байтов))
Как и следовало ожидать, команда QUIT закрывает соединение с сервером:
CLIENT: QUIT
SERVER: +OK dewey POP3 server signing off
CLIENT: QUIT
SERVER: +OK dewey POP3 server signing off (maildrop empty)
CLIENT: QUIT
SERVER: +OK dewey POP3 server signing off (2 messages left)
Обратите внимание на то, что отмеченные для удаления сообщения на самом деле не удаляются до тех пор, пока не выдана команда QUIT и не началась стадия обновления. В любой момент в течение сеанса клиент имеет возможность выдать команду RSET, и все отмеченные для удаления сообщения будут восстановлены.

Безопасность сети

Протоколы POP3 и SMTP не шифруются. Если кто-то получает доступ к сети, где работает сервер с запущенной службой POP3, почтовые сообщения пользователей потенциально могут быть прочитаны.

4. Конвергенция сетей.

Конвергенция информационных технологий - процесс сближения разнородных электронных технологий в результате их быстрого развития и взаимодействия.

Сближение локальных и глобальных сетей

Тесная интеграция локальных и глобальных сетей привела к значительному взаимопроникновению соответствующих технологий. Сближение в методах передачи данных происходит на платформе цифровой (немодулированной) передачи данных по волоконно-оптическим линиям связи. Процесс переноса технологий из глобальной сети Интернет в локальные приобрёл такой массовый характер, что появился термин intranet-технология.

Большой вклад в сближение локальных и глобальных сетей внесло доминирование протокола IP. Этот протокол сегодня используется поверх любых технологий локальных и глобальных сетей - Ethernet, Token Ring, ATM, frame relay - для создания из различных подсетей единой составной сети.

В локальных системах в последнее время большое внимание уделяется защите инфы от несанкционированного доступа.

Появились новые технологии, используемые для обоих видов сетей- ATM.

WORLD WIDE WEB- гипертекстовая информационная служба. Во многом превзошла аналогичные службы локальных сетей.

Одним из проявлений сближения локальных и глобальных сетей является появление сетей масштаба большого города, занимающих промежуточное положение между локальными и глобальными сетями. Городские сети, или сети мегаполисов (Metropolitan Area Networks, MAN), предназначены для обслуживания территории крупного города.

Конвергенция компьютерных и телекоммуникационных сетей

Их объединяет то, что в качестве ресурса выступает информация.

- Сближение видов услуг- IP- телефония ,услуги универсальной службы сообщений, объединяющей электронную почту, телеонию, факсимильную службу, пейджинговую связь.
- Технологическое сближение сетей происходит сегодня на основе цифровой передачи информации различного типа, метода коммутации пакетов и программирования услуг. Телефония давно сделала ряд шагов навстречу компьютерным сетям. Прежде всего, за счет представления голоса в цифровой форме, что делает принципиально возможным передачу телефонного и компьютерного трафика по одним и тем же цифровым каналам (телевидение также может сегодня передавать изображение в цифровой форме).

Дополнительные услуги телефонных сетей, такие как переадресация вызова, конференц-связь, телеголосование и др., могут создаваться с помощью так называемой интеллектуальной сети (Intelligent Network, IN), по своей сути являющейся компьютерной сетью с серверами, на которых программируется логика услуг.

Компьютерные сети также многое позаимствовали у телефонных и телевизионных сетей. Глобальные компьютерные сети строятся по такому же иерархическому принципу, что и телефонные, в соответствии с которым сети городов и районов объединяются в региональные сети, а те, в свою очередь, - в национальные и международные сети. Компьютерные сети берут на вооружение методы обеспечения отказоустойчивости телефонных сетей, за счет которых последние демонстрируют высокую степень надежности, так недостающую порой Интернету и корпоративным сетям.

Появился новый термин - инфокоммуникационпая сеть, который прямо говорит о двух составляющих современной сети - информационной (компьютерной) и телекоммуникационной.

5. Виды и характеристики физических каналов передачи данных

http://www.bmstu.ru/~iu/Vlasov/Pages/Page5_2.html

Линия связи (рис. 5.1) состоит в общем случае из физической среды, по которой передаются информационные сигналы, аппаратуры передачи данных и промежуточной аппаратуры. Синонимом термина линия связи (line) является термин канал связи (channel).

Физическая среда передачи данных (medium) может представлять собой кабель, то есть набор проводов, изоляционных и заоитных оболочек, соединительных разъемов, а также земную атмосферу или космическое пространство, через которые распространяются информационные сигналы.

В зависимости от среды передачи данных линии связи разделяются на:

- проводные (воздушные);
- кабельные (медные и волоконно-оптические);
- радиоканалы наземной и спутниковой связи.

Проводные линии

Проводные (воздушные) линии-провода, проложенные между столбами и висящие в воздухе. Скоростные качества и помехозащищенность этих линий оставляют желать много лучшего.

Кабельные линии.

Кабельные линии

Основные особенности конструкции кабелей схематично показаны на рис. 5.3.

- **Кабели на основе витой пары** называются симметричными кабелями из-за того, что они состоят из двух одинаковых в конструктивном отношении проводников. Симметричный кабель может быть как экранированным - на основе экранированной витой пары (Shielded Twisted Pair, STP), так и неэкранированным - на основе неэкранированной витой пары (Unshielded Twisted Pair, UTP). В первом случае кроме электрической изоляции проводящие жилы помещаются также внутрь электромагнитного экрана, в качестве которого чаще всего применяется проводящая медная оплетка.
- **Коаксиальный кабель (coaxial)** состоит из несимметричных пар проводников. Каждая пара представляет собой внутреннюю медную жилу и соосную с ней внешнюю жилу, которая может быть полой медной трубой или оплеткой, отделенной от внутренней жилы диэлектрической изоляцией. Внешняя жила играет двоякую роль - по ней передаются информационные сигналы, также она является экраном, защищающим внутреннюю жилу от внешних электромагнитных полей.

- **Волоконно-оптический кабель (optical fiber)** состоит из тонких (5-60 микрон) гибких стеклянных волокон (волоконных световодов), по которым распространяются световые сигналы. Это наиболее качественный тип кабеля - он обеспечивает передачу данных с очень высокой скоростью (до 10 Гбит/с и выше) и к тому же лучше других типов передающей среды обеспечивает защиту данных от внешних помех (в силу особенностей распространения света такие сигналы легко экранировать).

Каждый световод состоит из центрального проводника света (сердцевины) -стеклянного волокна, и стеклянной оболочки, обладающей меньшим показателем преломления, чем сердцевина.

Распространяясь по сердцевине, лучи света не выходят за ее пределы, отражаясь от покрывающего слоя оболочки. В зависимости от распределения показателя преломления и от величины диаметра сердечника различают:

- многомодовое волокно со ступенчатым изменением показателя преломления (рис. 5.4,а);
- многомодовое волокно с плавным изменением показателя преломления (рис. 5.4, б);
- одномодовое волокно (рис. 5.4, в).

Понятие **"мода"** описывает режим распространения световых лучей во внутреннем сердечнике кабеля. **В одномодовом кабеле (Single Mode Fiber, SMF)** используется центральный проводник очень малого диаметра, соизмеримого с длиной волны света - от 5 до 10 мкм. При этом практически все лучи света распространяются вдоль оптической оси световода, не отражаясь от внешнего проводника. Сверхтонкий- сложно изгтовить.

В многомодовых кабелях (Multi Mode Fiber, MMF) используются более широкие внутренние сердечники, которые легче изготовить технологически. В стандартах определены два наиболее употребительных многомодовых кабеля: 62,5/125 мкм и 50/125 мкм, где 62,5 мкм или 50 мкм - диаметр центрального проводника, а 125 мкм - диаметр внешнего проводника.

В многомодовых кабелях во внутреннем проводнике одновременно существует несколько световых лучей, отражающихся от внешнего проводника под разными углами. Угол отражения луча называется модой луча. Интерференции лучей разных мод ухудшает качество передаваемого сигнала, что приводит к искажениям передаваемых импульсов в многомодовом оптическом волокне. Характеристики существенно хуже, чем одномодовых. В результате многомодовые кабели используются в основном для передачи данных на небольшие расстояния (до 300-2000 м) на скоростях не более 1 Гбит/с, а одномодовые - для передачи данных со сверхвысокими скоростями в несколько десятков гигабит в секунду (а при использовании технологии DWDM - до нескольких терабит в секунду), на расстояниях от нескольких километров (локальные и городские сети) до нескольких десятков и даже сотен километров (дальняя связь).

Радиоканалы наземной и спутниковой связи

Радиоканалы наземной и спутниковой связи образуются с помощью передатчика и приемника радиоволн. Существует большое разнообразие типов радиоканалов, отличающихся как используемым частотным диапазоном, так и дальностью канала. Диапазоны коротких, средних и длинных волн (КВ, СВ и ДВ), называемые также диапазонами амплитудной модуляции (Amplitude Modulation, AM) по типу используемого в них метода модуляции сигнала, обеспечивают дальнюю связь, но при невысокой скорости передачи данных. Более скоростными являются каналы, работающие на диапазонах ультракоротких волн (УКВ), для которых характерна частотная модуляция (Frequency Modulation, FM), а также диапазонах сверхвысоких частот (СВЧ, или microwaves). В диапазоне СВЧ (свыше 4 ГГц) сигналы уже не отражаются ионосферой Земли, и для устойчивой связи требуется наличие прямой видимости между передатчиком и приемником. Поэтому такие частоты используют либо спутниковые каналы, либо радиорелейные каналы, где это условие выполняется.

6. Сети Frame Relay.

Протокол Frame Relay (I.122 ITU-t; ANSI T1S1.2; RFC-1490, -1315, -1604; см. также www.frforum.com/frame-relay/5000/approved/frf.3/frf.3.1/frf3.f.0.html) является одним из новых телекоммуникационных протоколов (1993 г.), он обеспечивает большую скорость передачи данных (1,5Мбит/с), меньшие задержки, но и меньшую надежность доставки информации. Frame Relay предназначен для межсетевого общения, ориентирован на соединение и использует два протокольных уровня модели OSI. Остальные уровни должны реализоваться программно. Такая схема заметно удешевляет интерфейс. Протокол вводит понятие committed information rates (CIR - оговоренные скорости передачи), обеспечивая каждому приложению гарантированную полосу пропускания. Если приложение не использует полностью выделенную полосу, другие приложения могут поделить между собой свободный ресурс. Frame Relay гарантирует большее быстродействие, чем X.25. Стандарт предусматривает 2-х, 3-х и 4-х байтовые форматы заголовков (ANSI T1.618 и ITU-T Q.922) и синхронную передачу данных. Применение инкапсуляции гарантирует транспортировку пакетов других протоколов через сети Frame Relay. Пакет Frame Relay начинается и завершается разграничительным байтом 0x7e (что соответствует и стандарту X.25). Максимальный размер кадра 1600 октетов. Формат пакета показан на рис. 4.3.4.1.

Рис. 4.3.4.1. Формат пакетов Frame Relay (цифры сверху - номера байт)

NLPID - идентификатор протокола сетевого уровня (network layer protocol ID).
FCS - двухбайтовая контрольная сумма кадра (frame control sum). Заполнитель является опционным и может отсутствовать.
Различные форматы заголовков кадров Frame Relay показаны на рисунках 4.3.4.2, 4.3.4.3 и 4.3.4.4. В верхней части рисунка приведена нумерация бит.

Рис. 4.3.4.2. 2-байтовый заголовок пакета Frame Relay (адрес)

C/R бит *command/response* (Команда/Отклик).

E/A бит *extended address* (Расширенный адрес) определяет, следует ли рассматривать следующий байт в качестве части адреса (E/A=0 заголовок продолжается в следующем октете).

DLCI (data link control interface) адрес управляющего интерфейса информационного канала (имеет только локальный смысл). В двухбайтовой версии DLCI занимает в сумме 10 бит.

FECN бит *forward explicit congestion notification* (указание на возможность реагирования на перегрузку при посылке пакетов). Сигнализирует отправителю о переполнении буферов на приеме.

BECN бит *backward explicit congestion notification* (тоже для случая приема пакетов).

DE бит *discard eligibility* (пометка пакета при перегрузке канала). Помеченный пакет может быть отброшен и потребуется его повторная пересылка.

При возникновении перегрузки DCE-узел отправляет устройствам-адресатам пакет с FECN=1, а узлам, шлющим ему информацию, пакет с битом BECN=1. Большое число пакетов с такими битами говорит о перегрузке и отправитель должен снизить частоту посылки пакетов или вовсе ее прекратить.

Рис. 4.3.4.3. 3-байтовый заголовок пакета Frame Relay

D/C - бит data/control (данные/управление) определяет, является ли последующее поле младшей частью DLCI или его следует интерпретировать как управляющую информацию DL-core.

Рис. 4.3.4.4. 4-байтовый заголовок пакета Frame Relay

Первым передается младший бит байта. Для управления сетью используется протокол snmp и база данных MIB. Формат кадра Frame Relay показан на рис. 4.3.4.4.

NLPID - (network layer protocol identifier) идентификатор протокола сетевого уровня. Это поле может содержать коды многих протоколов, включая IP, CCITT Q.933, ISO 8208, IEEE SNAP, CLNP (ISO 8473) и т.д. Это поле говорит получателю, какой тип протокола инкапсулирован. Коды nlpid стандартизованы документом ISO/IEC TR 9577. Некоторые допустимые коды этого поля приведены в таблице 4.3.4.1. Пользовательская информация располагается, начиная с поля управления, и содержит код 0x03 для случая пересылки без подтверждения (Q.922, UI). Для всех прочих видов обмена (кадры I- S-типов) подтверждение доставки является обязательным. Поле заполнитель предназначено для выравнивания границы полей на 2-байтовый уровень. Длина этого поля может быть равной нулю или одному байту. Поле адрес описано выше (см. рис. 4.3.4.1, 4.3.4.2, 4.3.4.3). Если за кодом NLPID следует 4 октета уровней 2 и 3, это указывает на то, что используется связь, ориентированная на соединение. Протокол Frame Relay предусматривает гибкую систему межсетевых соединения на основе мостов-шлюзов и маршрутизаторов. Все мосты и маршрутизаторы должны быть способны воспринимать и правильно интерпретировать как NLPID- так и SNAP-инкапсуляцию. Для обеспечения правильной интерпретации идентификатора протокола PID, предусмотрен 3-октетный уникальный идентификатор OUI (organizationally unique identifier). В пакетах для мостов и маршрутизатором в поле OUI предшествует двух-октетному полю PID.

Рис. 4.3.4.5. Формат маршрутизуемого кадра Frame Relay

Нетрудно видеть, что кадр Frame Relay имеет много общего с X.25 и ISDN. Здесь уже на протокольном уровне предусматривается мультикастинг.

Таблица 4.3.4.1. Коды поля NLPID (идентификатор протокола сетевого уровня)

Тип кадра	Название протокола	Код
I-кадр (ISO 8208)	N по модулю 8 N по модулю 128	0x0 1

		0x10
UI-кадр	IP clnp Q.933 SNAP Q.922 802.2 Протокол, заданный пользователем (уровень 3)	0xcc 0x81 0x08 0x80 0x4e 0x4c 0x70

Код протокола SNAP используется и для протоколов 802.3, 802.4, 802.5, FDDI и 802.6. При вложении IP в кадры Frame Relay в поле управления записывается код 0x03, а в поле NLPID - 0xcc, начиная с байта 5 располагается тело IP-дейтограммы, за которой следует поле FCS. Формат маршрутизируемой IP-дейтограммы показан на рис. 4.3.4.5А.

Рис. 4.3.4.5А. Формат маршрутизируемой IP-дейтограммы

Аналогично осуществляется инкапсуляция пакетов протокола clnp, только здесь в поле NLPID записывается код 0x81. Для примера на рис. 4.3.4.6 и 4.3.4.7 показаны пакеты для мостов 802.3 и FDDI (см. [“Multiprotocol Encapsulation over Frame Relay”](#)).

Рис. 4.3.4.6 Формат мостового кадра Ethernet 802.3

Рис. 4.3.4.7 Формат мостового кадра FDDI

7. Маршрутизация: маршрутизация первого уровня.

Функциональное соответствие видов коммуникационного оборудования уровням модели OSI

Лучшим способом для понимания отличий между сетевыми адаптерами, повторителями, мостами/коммутаторами и маршрутизаторами является рассмотрение их работы в терминах модели OSI. Соотношение между функциями этих устройств и уровнями модели OSI показано на рисунке 2.

Повторитель, который регенерирует сигналы, за счет чего позволяет увеличивать длину сети, работает на физическом уровне.

Сетевой адаптер работает на физическом и канальном уровнях. К физическому уровню относится та часть функций сетевого адаптера, которая связана с приемом и передачей сигналов по линии связи, а получение доступа к разделяемой среде передачи, распознавание MAC-адреса компьютера - это уже функция канального уровня.

Мосты выполняют большую часть своей работы на канальном уровне. Для них сеть представляется набором MAC-адресов устройств. Они извлекают эти адреса из заголовков, добавленных к пакетам на канальном уровне, и используют их во время обработки пакетов для принятия решения о том, на какой порт отправить тот или иной пакет. Мосты не имеют доступа к информации об адресах сетей, относящейся к более высокому уровню. Поэтому они ограничены в принятии решений о возможных путях или маршрутах перемещения пакетов по сети.

Маршрутизаторы работают на сетевом уровне модели OSI . Для маршрутизаторов сеть - это набор сетевых адресов устройств и множество сетевых путей. Маршрутизаторы анализируют все возможные пути между любыми двумя узлами сети и выбирают самый короткий из них. При выборе могут приниматься во внимание и другие факторы, например, состояние промежуточных узлов и линий связи, пропускная способность линий или стоимость передачи данных.

Для того, чтобы маршрутизатор мог выполнять возложенные на него функции ему должна быть доступна более развернутая информация о сети, нежели та, которая доступна мосту. В заголовке пакета сетевого уровня кроме сетевого адреса имеются данные, например, о критерии, который должен быть использован при выборе маршрута, о времени жизни пакета в сети, о том, какому протоколу верхнего уровня принадлежит пакет.

Благодаря использованию дополнительной информации, маршрутизатор может осуществлять больше операций с пакетами, чем мост/коммутатор. Поэтому программное обеспечение, необходимое для работы маршрутизатора, является более сложным. На рисунке 2 показан еще один тип коммуникационных устройств - шлюз, который может работать на любом уровне модели OSI. Шлюз (gateway) - это устройство, выполняющее трансляцию протоколов. Шлюз размещается между взаимодействующими сетями и служит посредником, переводящим сообщения, поступающие из одной сети, в формат другой сети. Шлюз может быть реализован как чисто программными средствами, установленными на обычном компьютере, так и на базе специализированного компьютера. Трансляция одного стека протоколов в другой представляет собой сложную интеллектуальную задачу, требующую максимально полной информации о сети, поэтому шлюз использует заголовки всех транслируемых протоколов.

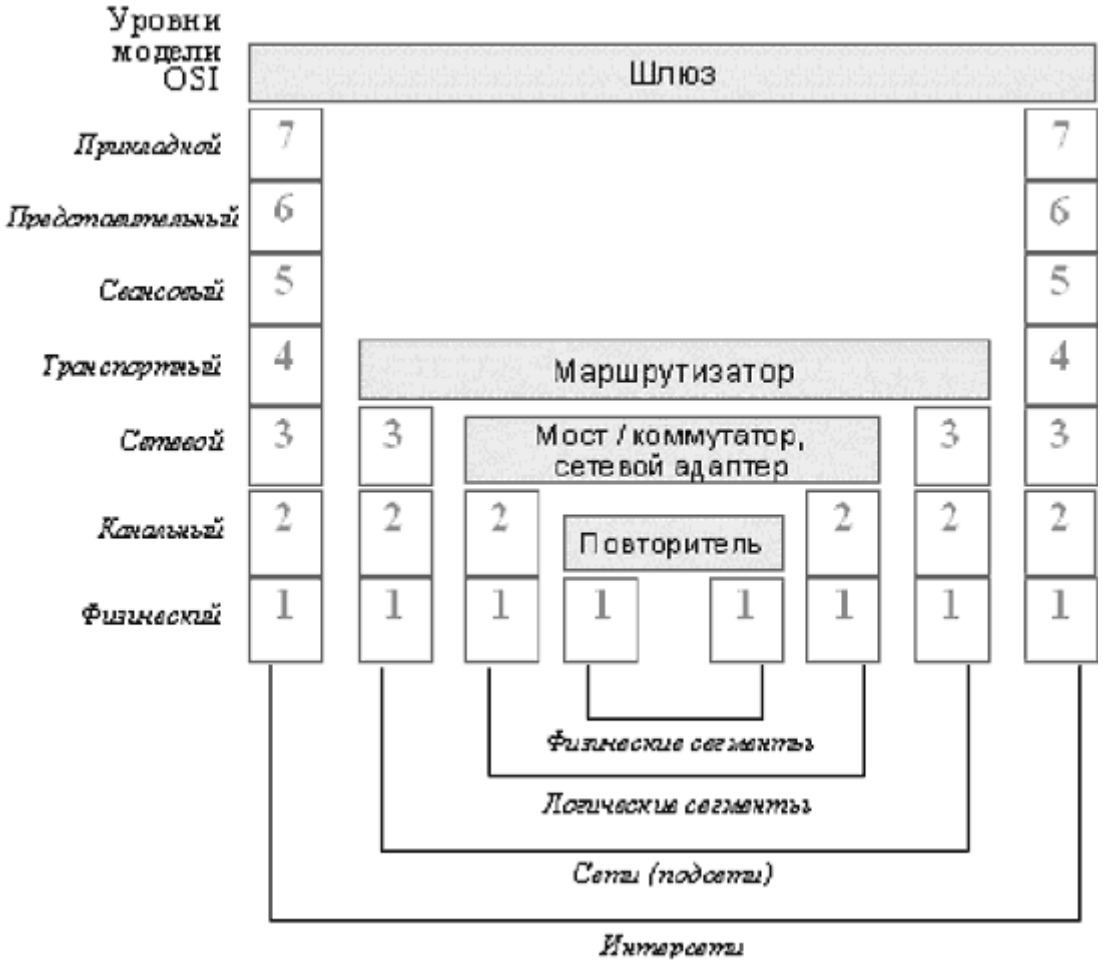


Рис. 2. Соответствие функций коммуникационного оборудования модели OSI

8. Сети ATM.

В настоящее время начинают широко внедряться каналы с пропускной способностью 150,52 и 622,08 Мбит/с. Эти каналы как для соединения локальных сетей, так и непосредственно для построения скоростных LAN. 150 Мбит/с может обеспечить любые современные телекоммуникационные услуги кроме телевидения высокого разрешения. Предусмотрен стандарт и на скорость передачи 2,48832 Гбит/с. Так как время доставки для многих видов сетевых услуг реального времени является крайне важной характеристикой, ATM находит широкое применение в телефонии, кабельном телевидении и других областях. Следует учитывать, что оцифрованный видеосигнал качества VHS требует 100Мбит/с при отсутствии сжатия и 1,5-6 Мбит/с при использовании сжатия. Файл изображения 1000x1000 пикселей при 24 битах, характеризующих цвет, занимает 3 Мбайта. ATM справится с передачей такого кадра с учетом накладных расходов (заголовок) за ~0,2 сек. Понятно, что при использовании сжатия можно получить заметно большее быстродействие.

Это не значит, что доступны лишь указанные скорости, интерфейсы позволяют мультиплексировать большое число каналов с самыми разными скоростями обмена. Но мультиплексирование на таких частотах представляет собой значительную проблему. Определенные трудности представляет то обстоятельство, что в ATM трудно реализовать обмен без установления соединения (аналог utp в Интернет)

Протокол ATM (asynchronous transfer mode; см. также А.Н. Назаров, М.В. Симонов. "ATM. Технология высокоскоростных сетей". ЭКО-Трендз, М. 1998) является широкополос-

ной версией ISDN, работает на скорости 150,52 Мбит/с с пакетом постоянной длины и минимальным заголовком. Слово асинхронный в названии означает, что тактовые генераторы передатчика и приемника не синхронизованы, а сами ячейки передаются и мультиплексируются по запросам. При мультиплексировании используется статистическая технология. Асинхронная передача не предполагает упорядочивания ячеек по каналам при пересылке. ATM поддерживает аппаратную и пакетную коммутацию.

Каждый пакет ATM имеет 53 байта (в англоязычной документации пакеты ATM носят название cell (ячейка), этот термин введен, чтобы отличить пакеты ATM от пакетов низкоскоростных каналов), из них 48 байт несут полезную информацию (что для случая передачи звука, соответствует 6 мс). Для выделения пакета из потока используются такие же, как в ISDN разделительные байты (0x7E). Заголовок пакета содержит лишь 5 байт и предназначен главным образом для того, чтобы определить принадлежит ли данный пакет определенному виртуальному каналу. Отсутствие контроля ошибок и повторной передачи на физическом уровне приводит к эффекту размножения ошибок. Если происходит ошибка в поле идентификатора виртуального пути или виртуального канала, то коммутатор может отправить ячейку другому получателю. Таким образом, один получатель не получит ячейку, а другой получит то, что ему не предназначалось.

Виртуальный канал в ATM формируется также как и в ISDN. Формально эта процедура не является частью ATM-протокола. Сначала здесь формируется сигнальная схема, для этого посылается запрос с VPI=0 и VCI=5. Если процедура завершилась успешно, можно начинать формирование виртуального канала. При создании канала могут использоваться 6 разновидностей сообщений.

- **setup** - запрос формирования канала.
- **call proceeding** - запрос в процессе исполнения.
- **connect** - запрос принят.
- **connect ACK** - подтверждение получения запроса.
- **release** - сообщение о завершении.
- **release compleate** - подтверждение получения сообщения release.

Схема обмена сообщениями при установлении (и разрыве) виртуального соединения показана на рис. 4.3.5.1. Предполагается, что между ЭВМ-инициализатором и ЭВМ-адресатом находится два ATM-переключателя. Каждый из узлов по пути к месту назначения при получении запроса setup откликается, посылая сообщение call proceeding. Адрес места назначения указывается в сообщении setup. В ATM используется три вида адресов. Первый - имеет 20 байт и имеет структуру OSI-адреса. Первый байт указывает на вид адреса (один из трех). Байты 2 и 3 указывают на принадлежность стране, а байт 4 задает формат последующей части кода адреса, которая содержит 3 байта кода администрации (authority), 2 байта домена, 2 байта области и 6 байтов собственно адреса. Во втором формате байты 2 и 3 выделены для международных организаций, а не стран. Остальная часть адреса имеет тот же формат, что и в варианте 1. Третий формат является старой формой (CCITT E.164) 15-цифровых десятичных телефонных номеров ISDN. В ATM не специфицировано никакого алгоритма маршрутизации. Для выбора маршрута (от коммутатора к коммутатору) используется поле VCP. VCI используется лишь на последнем шаге, когда ячейка посылается от переключателя к ЭВМ. Такой подход упрощает маршрутизацию отдельных ячеек, так как при этом анализируется 12- а не 28-битовые коды. В каждом коммутаторе (переключателе) формируются специальные таблицы, которые решают проблему переадресации ячеек.

Следует обратить внимание на то, что виртуальный канал (circuit) и виртуальный проход (path) в данном контексте не тождественны. Виртуальный проход (маршрут) может содер-

жать несколько виртуальных каналов. Виртуальные каналы всегда являются полностью дуплексными.

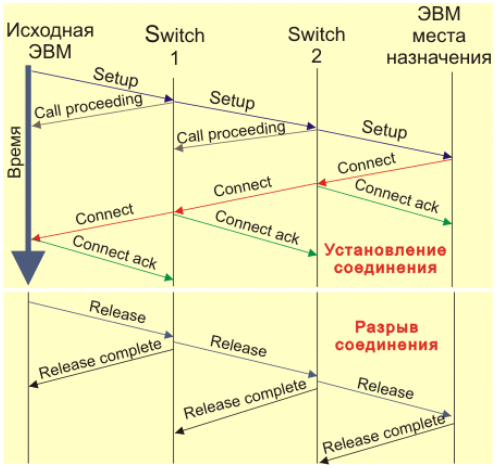


Рис. 4.3.5.1. Обмен сообщениями при установлении и разрыве виртуального соединения

Сети ATM допускают создание мультикастных каналов. Такой канал имеет одного отправителя и много получателей. Первый канал формируется обычным путем, последующие участники сессии подключаются позднее путем посылки сообщения add party.

За видимую простоту ячеек приходится платить тем, что управляющая информация передается в общем информационном потоке. Высокая скорость передачи данных требует применения аппаратно реализованных маршрутных таблиц на каждом переключателе пакетов. На рис. 4.3.5.2 представлен формат заголовка пакета ATM. Заголовок обеспечивает два механизма маршрутизации пакетов:

- **VPI** (virtual path identifier - виртуальный идентификатор маршрута) обеспечивает соединение точка-точка, но маршрут не является фиксированным и задается непосредственно перед началом пересылки с использованием сигнальных сообщений. Слово “виртуальный” означает, что пакеты передаются от узла к узлу в соответствии с VPI.

- **VCI** (virtual call identifier - виртуальный идентификатор запроса) запросы осуществляются в соответствии с виртуальным маршрутом, заданным VPI.

Эти два субполя вместе образуют поле маршрута, которое занимает 24 бита.

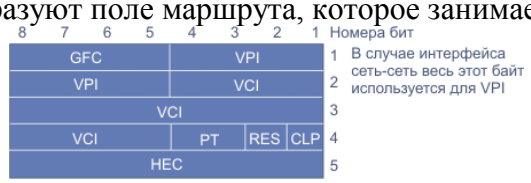


Рис. 4.3.5.2 Формат заголовка ATM-пакета (сетевой интерфейс пользователя - UNI)

Для интерфейса сеть-сеть (NNI) используется ячейка с несколько иным форматом заголовка. Там весь первый октет выделен для VPI, а поле GFC отсутствует.

GFC Generic flow control (4 бита, смотри описание пакетов ISDN) - общее управление потоком.

VPI Virtual path identifier (8 бит, служит для целей маршрутизации) - идентификатор виртуального

VCI Virtual call identifier (16 бит, служит для целей маршрутизации) - идентификатор виртуального

PT Payload type (2 бита, тип данных; это поле может занимать и зарезервированное субполе RES.)

RES зарезервированный бит.

CLP (Cell loss priority - уровень приоритета при потере пакета) указывает на то, какой приоритет им (cell), и будет ли он отброшен в случае перегрузки канала.

HEC header error control (8 бит, поле контроля ошибок)

Ряд значений VCI и VPI имеют фиксированные значения, приведенные в таблице 4.3.5.1

Таблица 4.3.5.1.

VCI	VPI	Назначение
0	только 0	Неопределенная ячейка
1	все	Мета управление
3	все	Сетевое управление VP-каналом
4	все	vp-управление для соединения между конечными точками
5	все	Управление доступом по схеме точка-точка
6	все	Ячейка управления ресурсами (для подавления перегрузки)
16	только 0	UNI (snmp) управление сетью

Некоторые значения поля pt зафиксированы, их значения представлены в таблице 4.3.5.2.

Таблица 4.3.5.2. Заданные значения поля PT (payload type identifier)

PT	Назначение ячейки	Взаимодействие пользователь-пользователь
000	Пользовательские данные (перегрузка отсутствует)	Нет
001	Пользовательские данные (перегрузка отсутствует)	Нет
010	Пользовательские данные (имеет место перегрузка)	Да
011	Пользовательские данные (имеет место перегрузка)	Да
100	Ячейка виртуального канала oam сегментного потока f5	
101	Соединение точка-точка oam сегментного потока f5	
110	Управление ресурсами	
111	Зарезервировано	

OAM - эксплуатация и техническое обслуживание. ATM обеспечивает любые услуги в сети:

- Передача голоса на скоростях 64 Кбит/с. Один ATM-пакет соответствует 6 мсек.
- Передача музыки с использованием схемы кодирования MUSICAM.
- Так как для случая изображения передается только переменная часть картинки, atm идеально подходит для решения такого рода задач.
- Задачи управления решаются менее экономно, но, тем не менее, достаточно эффективно (предусмотрено несколько приоритетов для управления потоками данных).

В ATM предусмотрено несколько категорий услуг (таблица 4.3.5.3).

Таблица 4.3.5.3. Типы категорий ATM-услуг

Класс	Описание	Пример
cbr	Постоянная скорость передачи	Канал T1
rt-vbr	Переменная скорость передачи (реальное время)	Видеоконференции
nrt-vbr	Пременная скорость передачи (нереальное вре-	Мультимедиа по электронной

	мя)	почте
abr	Доступная скорость передачи	Просмотр web-информации
ubr	Не специфицированная скорость передачи	Пересылка файлов в фоновом режиме

CBR не предусматривает контроля ошибок, управления трафиком или какой-либо другой обработки. Класс CBR пригоден для работы с мультимедиа реального времени.
Класс VBR содержит в себе два подкласса - обычный и для реального времени (см. таблицу выше). ATM в процессе доставки не вносит никакого разброса ячеек по времени. Случаи потери ячеек игнорируются.

Класс ABR предназначен для работы в условиях мгновенных вариаций трафика. Система гарантирует некоторую пропускную способность, но в течение короткого времени может выдержать и большую нагрузку. Этот класс предусматривает наличие обратной связи между приемником и отправителем, которая позволяет понизить загрузку канала, если это необходимо.

Класс UBR хорошо пригоден для посылки IP-пакетов (нет гарантии доставки и в случае перегрузки неизбежны потери).

atm использует исключительно модель с установлением соединения (здесь нет аналогий с UDP-протоколом). Это создает определенные трудности для управления трафиком с целью обеспечения требуемого качества обслуживания (QoS). Для решения этой задачи используется алгоритм GCRA (generic rate algorithm). Работа этого алгоритма проиллюстрирована на рис. 4.3.5.3.

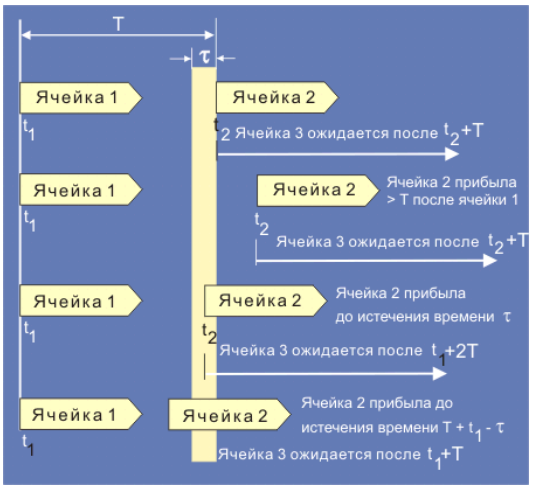


Рис. 4.3.5.3. Иллюстрация работы алгоритма GCRA

gcra имеет два параметра. Один из них характеризует максимально допустимую скорость передачи (PCR - peak cell rate; T=1/PCR - минимальное расстояние между ячейками), другой - допустимую вариацию значения скорости передачи (CDVT=L). Если клиент не собирается посылать более 100000 ячеек в секунду, то T=10 мксек. На рис. 4.3.5.3 представлены разные варианты следования ячеек. Если ячейка приходит раньше чем T-t, она считается неподтверждаемой и может быть отброшена. Ячейка может быть сохранена, но при этом должен быть установлен бит CLP=1. Применение бита CLP может быть разной для разных категорий услуг (смотри таблицу 4.3.5.3.). Данный механизм управления трафиком сходен с алгоритмом "дырявое ведро", описанным в разделе "Сети передачи данных".

Можно вычислить число подтверждаемых ячеек N, которые могут быть переданы при пиковом потоке ячеек PCR=1/t. Пусть время ячейки в пути равно d. Тогда N = 1 + (l/(T-d)). Если полученное число оказалось нецелым, оно должно быть округлено до ближайшего меньшего целого.

Трудно устранимой проблемой для atm является предотвращение перегрузки на промежуточных коммутаторах-переключателях. Коммутаторы могут иметь 100 внешних каналов, а загрузка может достигать 350000 ячеек/сек. Здесь можно рассматривать две задачи: подавление долговременных перегрузок, когда поток ячеек превосходит имеющиеся возможности их обработки, и кратковременные пиковые загрузки. Эти проблемы решаются различными способами: административный контроль, резервирование ресурсов и управление перегрузкой, привязанное к уровню трафика.

В низкоскоростных сетях с относительно медленно меняющейся или постоянной загрузке администратор вмешивается лишь при возникновении критической ситуации и предпринимает меры для понижения скорости передачи. Очень часто такой подход не слишком эффективен, так как за время доставки управляющих команд приходят многие тысячи ячеек. Кроме того, многие источники ячеек в ATM работают с фиксированной скоростью передачи (например, видеоконференция). Требование понизить скорость передачи здесь достаточно бессмысленно. По этой причине в ATM разумнее предотвращать перегрузку. Но для трафика типа CBR, VBR и UBR не существует никакого динамического управления перегрузкой и административное управление является единственной возможностью. Когда ЭВМ желает установить новый виртуальный канал, она должна охарактеризовать ожидаемый трафик. Сеть анализирует возможность обработки дополнительного трафика с учетом различных маршрутов. Если реализовать дополнительный трафик нельзя, запрос аннулируется. В отсутствии административного контроля несколько широкополосных пользователей могут блокировать работу массы узкополосных клиентов сети, например, читающих свою почту.
Резервирование ресурсов по своей сути близко административному контролю и выполняется на фазе формирования виртуального канала. Резервирование производится вдоль всего маршрута (во всех коммутаторах) в ходе реализации процедуры setup. Параметрами резервирования может быть значение пикового значения полосы пропускания и/или средняя загрузка.

Для типов сервиса CBR и VBR отправитель даже в случае перегрузки не может понизить уровень трафика. В случае UBR потери не играют никакой роли. Но сервис ABR допускает регулирование трафика. Более того, такое управление здесь весьма эффективно. Существует несколько механизмов реализации такого управления. Так предлагалось, чтобы отправитель, желающий послать блок данных, сначала посылал специальную ячейку, резервирующую требуемую полосу пропускания. После получения подтверждения блок данных начинает пересылаться. Преимуществом данного способа следует считать то, что перегрузки вообще не возникает. Но данное решение не используется из-за больших задержек (решение ATM-форума).

Другой способ сопряжен с посылкой коммутаторами специальных ячеек отправителю в случае возникновения условий перегрузки. При получении такой ячейки отправитель должен понизить скорость передачи вдвое. Предложены различные алгоритмы последующего восстановления скорости передачи. Но и эта схема отвергнута форумом atm из-за того, что сигнальные ячейки могут быть потеряны при перегрузке. Действительно данный алгоритм не всегда можно признать разумным. Например, в случае, когда коммутатор имеет 10 каналов с трафиком по 50 Мбит/с и один канал с потоком в 100 кбит/с, глупо требовать понижения трафика в этом канале из-за перегрузки.

Третье предложение использует тот факт, что граница пакета помечается битом в последней ячейке. Коммутатор просматривает входящий поток и ищет конец пакета, после чего выбрасывает все ячейки, относящиеся к следующему пакету. Этот пакет будет переслан позднее, а отбрасывание M ячеек случайным образом может вынудить повторение передачи m пакетов, что значительно хуже. Данный вариант подавления перегрузки был также не принят, так как выброшенный пакет совсем не обязательно послан источником, вызвавшим перегрузку. Но этот способ может быть использован отдельными производителями коммутаторов.

Обсуждались решения, сходные с тем, что используется в протоколе TCP "скользящее окно". Это решение требует слишком большого числа буферов в коммутаторах (как минимум по одному для каждого виртуального канала). После длинных дискуссий был принят за основу совершенно другой метод.

После каждых M информационных ячеек каждый отправитель посылает специальную RM-ячейку (resource management). Эта ячейка движется по тому же маршруту, что и информационные, но RM-ячейка обрабатывается всеми коммутаторами вдоль пути. Когда она достигает места назначения, ее содержимое просматривается и корректируется, после чего ячейка посылается назад отправителю. При этом появляются два дополнительных механизма управления перегрузкой. Во-первых, RM-ячейки могут посылаться не только первичным отправителем, но и перегруженными коммутаторами в направлении перегрузившего их отправителя. Во-вторых, перегруженные коммутаторы могут устанавливать средний PTI-бит в информационных ячейках, движущихся от первоисточника к адресату. Но даже выбранный метод подавления перегрузки не идеален, так как также уязвим из-за потерь управляющих ячеек.

Управление перегрузкой для услуг типа abr базируется на том, что каждый отправитель имеет текущую скорость передачи (ACR - actual cell rate), которая лежит между MCR (minimum cell rate) и PCR (peak cell rate). Когда происходит перегрузка, ACR уменьшается, но не ниже MCR. При исчезновении перегрузки acr увеличивается, но не выше PCR. Каждая RM-ячейка содержит значение загрузки, которую намеривается реализовать отправитель. Это значение называется ER (explicit rate). По пути к месту назначения эта величина может быть уменьшена попутными коммутаторами. Ни один из коммутаторов не может увеличивать ER. Модификация ER может производиться как по пути туда, так и обратно. При получении RM-ячейки отправитель может скорректировать значение ACR, если это необходимо.

С точки зрения построения интерфейса и точек доступа (T, S и R) сеть ATM сходна с ISDN (см. рис. 4.3.3.1).

Для физического уровня предусмотрены две скорости обмена 155,52 и 622,08 Мбит/с. Эти скорости соответствуют уровням иерархии SDH STM-1 и 4*STM-1. При номинальной скорости 155.52 Мбит/с пользователю доступна реально скорость обмена 135 Мбит/с, это связано с издержками на заголовки и управление. Для ATM используются коаксиальные кабели, скрученные пары (<100м для обоих вариантов) и оптоволоконные кабели (~2км). Для канала связи рассматриваются два кода CMI (coded mark inversion) и скрэмблеры типа установка-сброс (set-reset). В CMI двоичный 0 передается как отрицательный импульс половинной длины, за которым следует положительный импульс той же длительности. Двоичная 1 представляется в виде отрицательного или положительного импульса полной длины, так чтобы уровень менялся для последовательно следующих 1 (система AMI, см. раздел 2.1). Это обеспечивает балансировку передающей линии по постоянному напряжению, но удваивает частоту переключения практически вдвое. Скрамблерный метод не меняет частоту переключения, но его эффективность зависит от передаваемой информации. CMI предпочтительней для 155 Мбит/с. В настоящее время используется две схемы передачи данных применительно к ATM: базирующийся на потоке пакетов (cell stream) и на SDH структурах. В первом случае мы имеем непрерывный поток 53-октетных пакетов, во втором эти пакеты уложены в STM-1 кадры. Управляющие сообщения располагаются в заголовках секции и пути кадра SDH. AAL (ATM adaptation layer) служит для адаптации различных видов сервиса к требованиям ATM-уровня. Каждый вид услуг требует своего AAL-протокола. Главной целью AAL является обеспечение удобства при создании и исполнении программ прикладного уровня. Для всех AAL определены два субуровня:

SAR (segmentation and reassemble) делит пакеты высокого уровня, передает atm и наоборот (сборка сообщений из сегментов).

CS (convergent sub-layer) зависит от вида услуг (обработка случаев потери пакета, компенсация задержек, мониторирование ошибок и т.д.). Этот подуровень может в свою оче-

редь делиться на две секции: CPCS (common part convergence sublayer) - общая часть субуровня конвергенции и SSCS (service-specific convergence sublayer) - служебно-ориентированный подуровень конвергенции (последний может и отсутствовать).

AAL-протоколы управляются значениями следующих переменных:

- Скорость обмена (постоянная или переменная)
- Режим соединения (с установлением связи или без)
- Синхронизация (требуется или нет синхронизация между отправителем и получателем)

В настоящее время определены четыре класса услуг, которые могут требовать или нет синхронизации между отправителем и получателем, осуществлять обмен при постоянной или переменной частоте передачи бит, с установлением связи или без. Особенности этих видов услуг для адаптивного уровня систематизированы в таблице 4.3.5.4. Каждая из услуг имеет свой AAL протокол.

Таблица 4.3.5.4. Особенности видов услуг для адаптивного уровня

	Класс a (AAL1)	**Класс b (AAL2)**	**Класс c (AAL3/4 или 5)**	**Класс d (AAL3/4 или 5)**
Синхронизация работы отправителя и получателя	необходима	необходима	не нужна	не нужна
Частота следования битов	Постоянная	Переменная	Переменная	Переменная
Режим соединения	С соединением	С соединением	С соединением	Без соединения

Уровень адаптации 1-го уровня (AAL) выполняет для верхнего уровня следующие услуги (передача аудио- и видео- по каналам DS-1 и DS-3; постоянная скорость передачи):

- синхронизацию передатчика и приемника;
- передачу данных с фиксированной скоростью;
- индикацию потери и искажения данных, если эти ошибки не устраняются на уровне адаптации;
- передачу от отправителя получателю информации о структуре передаваемых данных.

Для решения этих задач AAL первого уровня должен устранять разброс задержек, выявлять ячейки, доставленные не по адресу, и потерянные ячейки, сегментацию пакетов и последующее их восстановление, выполнять мониторирование ошибок в управляющей информации протокола AAL-PCI (protocol control information). Характер обмена здесь строго ориентирован на соединение. AAL-1 использует субуровни конвергенции и SAR. Субуровень конвергенции обеспечивает постоянство скорости передачи ячеек. AAL-1 конвергенции не имеет какого-то специфического протокольного заголовка. Этот субуровень разбивает входные сообщения на 46- или 47-байтные блоки и передает их субуровню SAR для пересылки.
Структура протокольной части информационного поля ячейки SAR-pdu представлена на рис. 4.3.5.4
CSI позволяет приемнику распознать уровень конвергенции. Подуровень SAR получает значение SN (порядковый номер) для каждого 47-октетного блока данных от подуровня

конвергенции. Поле SNR (sequence number protection - контрольная сумма) служит для обнаружения и исправления ошибок в заголовке, в качестве производящего полинома используется G(x)= x3 + x + 1. Один из битов SNP- представляет собой бит четности. Если CSI=1, то после поля SNP следует однобайтовое поле указатель, которое используется для определения положения начала следующего сообщения (значения 0-92; старший бит поля указатель зарезервирован на будущее). Поле данных в этом варианте имеет 64 байт.
Для сжатой аудио и видео информации скорость передачи может варьироваться в широких пределах. Ведь многие схемы предусматривают периодическую отправку полного видеокадра при последующей передаче транспортируются лишь отличия последовательных кадров. Уровень адаптации 2-го типа предоставляет вышестоящему уровню возможность синхронизовать передатчик и приемник, осуществлять обмен с изменяющейся скоростью, оповещение об ошибках и потерях ячеек. Структура ячейки AAL 2-го типа показана на рис. 4.3.5.5 (субуровень SAR). Из-за переменной скорости передачи заполнение ячеек может быть неполным.

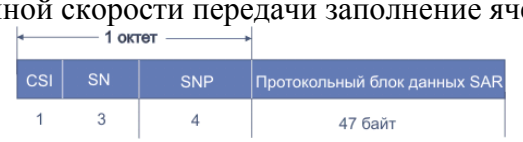


Рис. 4.3.5.4. Структура PDU подуровня SAR ATM 1-го типа (AAL1)

CSI (convergence sublayer indicator) - индикатор подуровня конвергенции

SN (sequence number) - номер по порядку

SNP (sequence number protection) - защита номера последовательности

Рис. 4.3.5.5. Структура PDU подуровня SAR ATM 2-го типа (AAL2)

IT (information type) - тип данных. Служит для указания начала, продолжения или окончания сооб

LI (length indicator) - индикатор длины. Указывает число октетов в поле данных

CRC Контрольная сумма

Поля SN и IT имеют общую длину 1 байт, поля же LI и CRC вместе занимают 2 байта. Поле данных (PDU) в такой ячейке имеет длину 45 байт. Более детальной информации о длинах полей стандарт не оговаривает.
Уровень адаптации 3/4 типов предназначен для передачи данных как в режиме с установлением соединения, так и без него. Раньше службам C и D были выделены разные типы уровня адаптации, позднее они были объединены. Определены два типа обмена: сообщение и поток. В первом случае блок данных передается в одном интерфейсном блоке (IDU). Сервисные блоки данных могут иметь переменную длину. В режиме поток сервисный блок данных передается через интерфейс уровня адаптации в одном или нескольких IDU. В этом режиме может быть реализована услуга “внутренний контейнер”. Здесь допускается и прерывание передачи, частично переданный блок теряется. AAL3/4 допускает организацию нескольких сессий одновременно (например, несколько удаленных login). Структура протокольного блока данных подуровня SAR 3/4 типа представлена на рис. 4.3.5.6. Длина поля данных (PDU) составляет 44 байта. Заметим, что AAL3/4 имеет два уровня издержек - 8 байт добавляется для каждого сообщения и 4 избыточных байта приходятся на каждую ячейку, это достаточно много особенно для коротких сообщений.

Рис. 4.3.5.6. Структура pdu подуровня SAR ATM 3/4-го типов

ST (segment type) - тип сегмента. Начало сообщения - 10 (BOM - beginning of message), продолжение - 00 (COM - continuation of message), завершение сообщения - 01 (EOM - end of message), односегментное сообщение - 11;

SN (sequence number) - номер по порядку;

MID (multiplexing identifier) - идентификатор мультиплексирования для протокола 4-го уровня (позволяет мультиплексировать до 1024 пользователей для одного соединения). Поле служит для определения того, к какой из активных сессий принадлежит данная ячейка.

li длина заполнения поля данных.

При вычислении crc используется образующий полином G(x) = x11 + x9 + x5 + x4 + x +1. Подуровень конвергенции aal содержит общую часть подуровня CPCS (common path convergence sublayer) и служебную часть подуровня SSCS (service specific convergence sublayer). CPCS обеспечивает негарантированную доставку кадров любой длины в диапазоне 1-65535 байт. Данные пользователя передаются непосредственно на субуровень AAL. Формат протокольного блока данных подуровня конвергенции AAL 3/4-типа показан на рис. 4.3.5.7.

Рис. 4.3.5.7. Формат блока данных подуровня конвергенции AAL 3/4-типа

CPI (common part indicator) - однооктетный индикатор общей части, используется при интерпретации последующих полей;

BTAG (beginning tag) - однооктетная метка начала, в сочетании с ETAG определяет границы протокольного блока данных (PDU);

BAsize (buffer allocation size) - емкость буфера, сообщает получателю максимальный размер буфера. Поле занимает 2 байта;

PAD заполнитель, обеспечивает кратность поля данных 4 октетам;

AL (alignment) - выравнивание, заполняется нулями;

ETAG (end tag) - метка конца (один октет);

Длина задает протяженность cpcs-pdu;

CPCS-PDU (common part convergence sublayer - protocol data unit) - протокольный блок данных общей части подуровня конвергенции

Тип 3/4 имеет существенную избыточность (4 байта из 48 на каждый SAR-PDU). По этой причине был введен 5-ый тип. Этот уровень обеспечивает канал, ориентированный на соединение, с переменной скоростью обмена (VBR) в широковещательном режиме при минимальном контроле ошибок (или вовсе без него). IP-дейтограммы передаются через сети ATM через адаптационный уровень 5 (RFC-1577). Уровень AAL5 иногда называют SEAL (simple and efficient adaptation layer - простой и эффективный адаптационный уровень). AAL5 занимает в наборе протоколов семейства ATM нишу протокола udp стека TCP/IP. Формат ячейки SAR-PDU 5-го типа показан на рис. 4.3.5.8.

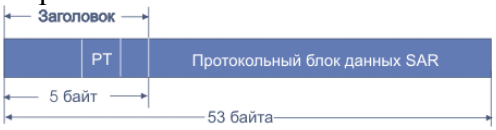


Рис. 4.3.5.8. Формат ячейки SAR-PDU 5-го типа (AAL5)

Рис. 4.3.5.8а. Формат сообщения AAL5 субуровня конвергенции

UU (user to user) - поле необходимо для верхних уровней, чтобы обеспечить мультиплексирование;

Длина двухоктетное поле длины поля данных (PDU);

CRC 4-октетная контрольная сумма;

Однобайтовое поле, расположенное между полями UU и длина зарезервировано для использования в будущем. Так как здесь для переноса информации используется заголовок, работа AAL не является независимой от нижележащего уровня, что является нарушением эталонной модели. Инкапсулироваться в поля данных AAL5 могут блоки длиной до 216-1 октетов (65535). Выполнение операций здесь зависит от того, работает ли система в режиме сообщения или потока. На подуровне конвергенции для передачи протокольного блока данных используется 4-х байтовая CRC с образующим полиномом G(x) = x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1, что обеспечивает высокую надежность корректности доставки. Положение адаптационного уровня в рамках эталонной модели показано на рис. 4.3.5.9. Следует впрочем заметить, что не вполне ясно, какой уровень занимает сам протокол ATM (транспортный или сетевой?).

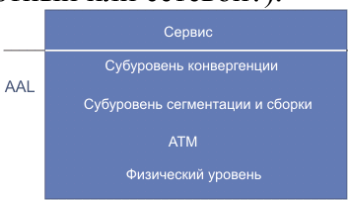


Рис. 4.3.5.9. Положение уровней ATM в универсальной модели

Верхние уровни управления для ATM базируются на рекомендациях ccitt I450/1 (Q.930/1). В случае использования ATM для Интернет значение MTU по умолчанию равно 9180 (RFC-1626), так как фрагментация IP-дейтограмм крайне нежелательна (AAL). Работа протоколов TCP/IP поверх ATM описана в документах RFC-1483, -1577, -1626, -1680, -1695, -1754, -1755, -1821, -1926, -1932 (полужирным шрифтом выделены коды документов, являющиеся стандартами Интернет). Ниже на рис. 4.3.5.10 показано, как пакеты atm размещаются в кадрах STM-1 (виртуальный контейнер VC-4).

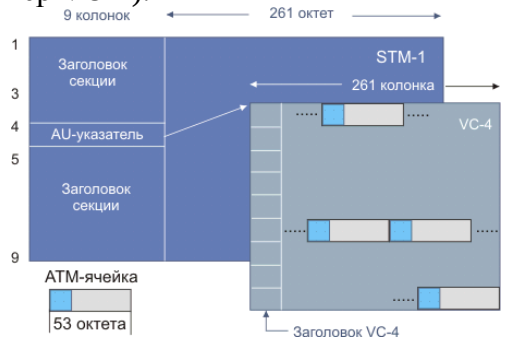


Рис. 4.3.5.10. Размещение atm пакетов в STM-1 кадре

В STM-1 для передачи ячеек выделяется полоса пропускания $\mathbf{I} = \frac{9 \times 261 \times 8}{125 \times 10^{-6}}$ =150,3 Мбит/с. (9 рядов по 261 байту, передаваемые каждые 125 мксек)

Форматы адресов согласно спецификации интерфейса “пользователь-сеть” представлены на рис. 4.3.5.11.

Рис. 4.3.5.11. Формат DCC ATM с числовым кодом страны.

AFI (authority and format identifier) - идентификатор формата и привилегий.

DCC (data country code) - код данных страны (стандарт МОС 3166).

DFI (DSP format identifier) - идентификатор формата DSP.

DSP (domain specific part) - часть, зависящая от домена.

AA (administrative authority) - административное субполе.

RSVD (reserved) - резерв на будущее.

RD (routing domain) - область маршрутизации.

AREA идентификатор зоны.

ESI (end system identifier) - идентификатор оконечной системы.

SEL (selector) - селектор.

IDI (initial domain identifier) - идентификатор исходной области.

HO (higher order) - старшая часть.

Формат ICD с указателем международного кода отличается от формата DCC тем, что в нем поле DCC заменено полем международного кода ICD (international code designator). Формат адреса E.164 NSAP, где идентификатор исходной области является номером E.164, представлен на рис. 4.3.5.12. Структура номера (15 десятичных цифр в кодировке BCD) места назначения отображена на рис. 4.3.5.13.

Важную роль в управлении сетями ATM играет информация OAM(operations and maintenance). Здесь осуществляется тесное взаимодействие с потоками управления sdh (F1-F5).

F1 - поток данных oam уровня регенерационной секции SDH.
F2 - поток данных oam цифровой мультиплексорной секции SDH.
F3 - поток данных oam уровня пути обмена SDH.
F4 - поток данных oam виртуального пути ATM.
F5 - поток данных oam виртуального канала ATM.

Рис. 4.3.5.12. Формат адреса E.164 NSAP

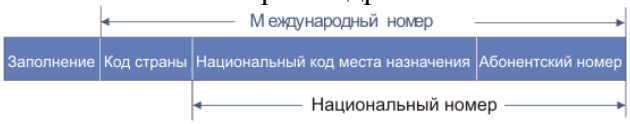


Рис. 4.3.5.13. Структура номеров

Код страны (CC -country code) занимает от одной до трех цифр (из 15).

Маршрутизация в atm отличается от аналогичных процессов в сетях с коммутацией пакетов. Сети ATM в основном ориентированы на соединение. Ячейки транспортируются по уже выбранному маршруту через коммутаторы ATM в соответствии со значениями идентификаторов виртуального пути и виртуального канала. Вычисление маршрута осуществляется на специальном сервере. Потоки информации F4 или F5 принимаются и обрабатываются устройствами, которые формируют виртуальные пути или каналы. Формат информационного поля ячейки oam показан на рис. 4.3.5.14. Поток информации oam F4 уровня виртуального пути для идентификации потока точка-точка использует идентификатор виртуального канала VCI=4, а для сегментных потоков VCI=3.

Рис. 4.3.5.14. Формат ячейки OAM F4

Поток ячеек OAM F5 уровня виртуального канала каких-либо специальных идентификаторов виртуальных путей не использует. В заголовках ячеек потока oam F5 типа точка-точка в поле типа данных (PT) записывается код 100, а для сегментных потоков виртуальных каналов PT=101. Значения кодов полей тип OAM и выполняемой функции приведены в таблице 4.3.5.5. Для решения проблем выявления и локализации отказов в сети ATM используются ячейки AIS (alarm indication signal - аварийный сигнал), RDI/FERF (remote defect indication/far end reporting failure - указатель отказа на удаленном конце), контроля непрерывности (continuity check) и проверки с применением обратной связи (loopback). Для ячеек AIS и RDI поля тип отказа имеет 8 байт (по умолчанию во все октеты записывается 0x6A), а для указа-

теля места отказа выделено 9 байт. Полезная часть поля данных в этих ячейках равна 45 байтам, из них 28 зарезервировано на будущее.

Таблица 4.3.5.5.

Код поля *тип oam*	**Назначение**	**Код поля *тип выполняемой функции***	**Назначение**
0001	Обнаружение и определение места отказов (fault management)	0000	Указание отказа (AIS)
		0001	Указание на удаленный дефект (RDI/FERF)
		0100	Проверка непрерывности (continuity check)
		1000	Обратная связь (loopback)
0010	Контроль рабочих характеристик	0000	Прямой мониторинг (forward monitoring)
		0001	Сообщение о предыстории (backward reporting)
		0010	Мониторирование и предоставление результатов (monitoring and reporting)
1000	Активизация и завершение процессов oam	0000	Мониторинг рабочих характеристик (performance monitoring)
		0001	Проверка непрерывности (continuity check)

Контроль рабочих характеристик сети АТМ производится без нарушения соединений и без снижения качества обслуживания. Для запуска и остановки процесса измерения служат ячейки типа activation/deactivation. В ячейке oam для этих целей в поле данных выделено 45 байт. Формат субполей поля данных представлен на рис. 4.3.5.15.

Рис. 4.3.5.15. Формат субполей поля данных oam activation/deactivation

Субполе неиспользуемые октеты заполняется байтами 0x6A, а субполя блок PM - кодами 0000. Значения кодов поля идентификатор сообщения приведены в таблице 4.3.5.6.

Таблица 4.3.5.6.

Код поля идентификатор сообщения	**Назначение**
000001	Активация (запрос)
000010	Подтверждение активации
000011	Отвержение запроса активации
000101	Деактивация
000110	Подтверждение деактивации
000111	>Отвержение запроса деактивации

В субполе направление действия заносится код 10 при направлении от А к В и 01 при противоположном направлении. В поле размер записывается код 1000 при длине 1024 ячеек, 0100 -

при 512, 0010 - при 256 и 0001 - при 128. Размеры блоков для направлений A ->b и B ->a могут быть и неравными. Мониторинг рабочих параметров может выполняться для A ->b, B ->a или для обоих направлений одновременно.

Формат ячеек oam типа измерение рабочих характеристик представлен на рис. 4.3.5.16.

Рис. 4.3.5.16. Формат ячеек oam типа измерение рабочих характеристик

В субполя BIP-16 (bit interleaved parity) и счет потерянных ячеек в отсутствии прямого мониторинга по умолчанию заносится код 0x6A, аналогичный код записывается в субполя число ячеек пользователя и результаты анализа в отсутствие обратного мониторинга. В неиспользуемое поле записываются 1 во все биты, если не использована временная метка. Поле порядковый номер мониторинга (MSN - monitoring sequence number) содержит номер ячейки oam типа PM по модулю 256. Поле общее число ячеек пользователя (TUC - total user cell) записывается число пользовательских ячеек, отправленных после последней ячейки OAM типа PM.

Один физический отказ может сгенерировать большое число ячеек OAM. Для блокировки такой возможности введено ограничение на период генерации таких ячеек (> нескольких секунд). Операции проверки тракта, выполняемые с помощью ячеек OAM типа loopback, позволяют выявить место возникновения неисправностей. Формат поля специальных функций ячейки OAM типа loopback отображен на рис. 4.3.5.17 (см. также рис. 4.3.5.14).

Индикатор шлейфа	Корреляционная метка	Идентификатор шлейфа	Идентификатор источника	Не использ.	0 или 1
8 бит	8 бит	12 байт	12 байт	135 бит	1 бит

Рис. 4.3.5.17. Формат ячейки oam типа loopback

Поле неиспользуемое содержит во всех октетах по умолчанию код 0x6A. Поле индикатор шлейфа содержит 1 при посылке отправителем (остальные семь бит равны нулю), единица заменяется нулем в момент приема. При получении ячейки с индикатором шлейфа, равным нулю, она уничтожается. Поле корреляционная метка используется отправителем для идентификации отклика. Поле идентификатор места шлейфа определяет место, откуда ячейка должна быть послана назад. Если поле содержит все единицы, таким местом является адресат. Поле идентификатор источника служит для распознавания ячейки и ее уничтожения при возвращении.

Предоставление услуг без установления соединения соответствует уровню выше чем ATM и требует соединения каждого клиента с соответствующим сервером, решающим данную задачу. Большинство локальных и региональных сетей ATM реализуют именно такой режим. Для передачи данных без установления соединения используется протокол доступа CLNAP (connectionless network access protocol), интерфейс CLAI (connectionless access interface) и сетевой протокол CLNIP (connectionless network interface protocol). Размер поля данных для CLNAP не является постоянным и составляет 9188 октетов, что подразумевает фрагментацию. Эти протоколы работают выше подуровня конвергенции. Соответствующая длина для CLNIP SDU равна 9236 октетам. Формат блока данных CLNIP показан на рис. 4.3.5.18.

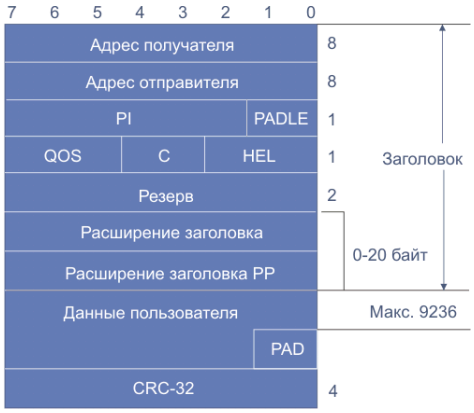


Рис. 4.3.5.18. Формат структуры данных протокола CLNIP

PI (protocol identifier) - идентификатор протокола.

PADLE (padding length) - длина заполнения.

QoS (quality of service) - качество обслуживания

C (CRC indication bit) - индикатор числа бит в контрольной сумме CRC.

HEL (header extension length) - длина расширения заголовка.

Проблему фрагментации и инкапсуляции этих длинных пакетов в ATM ячейки берет на себя коммутатор доступа. Схема вложения и фрагментации для пакетов clnap отображена на рис. 4.3.5.19.

Из рисунка видно, что на подуровне SAR происходит деление пакета на части и укладка полученных сегментов в поля данных ячеек (48 байт).

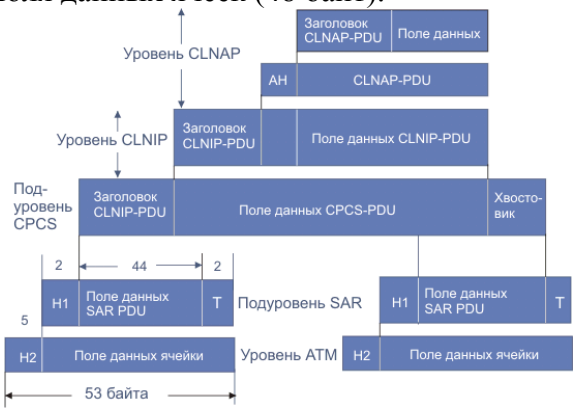


Рис. 4.3.5.19. Схема вложения и фрагментации для пакетов CLNAP

AH (alignment header; 4 октета) - поле выравнивания.

H2 Заголовок SAR-PDU.

H1 Заголовок ячейки.

T Хвостовик.

SAR (segmentation and reassemble) - сегментация и сборка.

CPCS (common part convergence sublayer) общая часть подуровня конвергенции.

Для использования одного и того же виртуального канала многими протоколами служит LLC-инкапсуляция (logical link control). LLC-заголовок укладывается в поле данных перед PDU и содержит в себе информацию, необходимую для того, чтобы корректно обработать AAL5 CPCS-PDU. Обычно такой заголовок имеет формат IEEE 802.2, за которым может следовать SNAP-заголовок IEEE 802.1a. LLC-заголовок, содержащий код 0xFE-fe-03, говорит о том, что далее следует маршрутизируемый pdu длиной 216-4 октетов. Одно-октетный код

NLPID идентифицирует сетевой протокол. Значения кодов NLPID представлены в таблице 4.3.5.7.

Таблица 4.3.5.7. Значения кодов NLPID

Код nlpid	**Назначение**
0x00	Нулевой сетевой уровень (в atm не используется)
0x80	SNAP
0x81	ISO CLNP
0x82	ISO ESIS
0x83	ISO ISIS
0xCC	Интернет (IP не является протоколом ISO)

Формат PDU для маршрутизируемых данных при использовании протоколов, не принадлежащих ISO, представлен на рис. 4.3.5.20 (случай IP-дейтограммы).

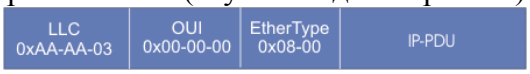


Рис. 4.3.5.20. Формат IP PDU при транспортировке с использованием AAL5 ATM

Пропускная способность сети ATM (150 Мбит/с) позволяет передавать немного более 360000 ячеек в секунду, что означает для ATM-переключателя время коммутации менее 2,7 мксек. Реальный переключатель может иметь от 16 до 1024 входных линий, что может означать коммутацию 16-1024 ячеек каждые 2,7 мксек. При быстродействии 622 Мбит/с новая порция ячеек поступает каждые 700 нсек. Постоянство длины ячеек упрощает конструкцию ключа. Все ATM-ключи имеют целью обеспечить коммутацию с минимальной вероятностью потери и исключить возможность изменения порядка следования ячеек. Приемлемой считается вероятность потери ячейки не более 10-12. Это эквивалентно для большого коммутатора потери 1-2 ячеек в час. Уменьшению вероятности потери способствует создание буферов конвейерного типа. Если на вход переключателя приходят две ячейки одновременно, одна из них обслуживается, а вторая ставится в очередь (запоминается в буфере). Выбор ячеек может производиться псевдослучайно или циклически. При этом не должно возникать предпочтений для каких-то каналов. Если в один цикл на вход (каналы 1, 2, 3 и 4) коммутатора пришли четыре ячейки, предназначенные для выходных линий J+2, J, J+2 и J+1, соответственно, то на линии J+2 возникает конфликт. Предположим, что будет обслужена ячейка, поступившая по первой входной линии, а ячейка на входной линии 3 будет поставлена в очередь. В начале следующего цикла на выход попадут три ячейки. Предположим также, что в этот цикл на ходы коммутатора (1 и 3) придут ячейки адресованные для линий J+3 и J, соответственно. Ячейка, адресованная J, будет поставлена в очередь вслед за ячейкой, адресованной J+2. Все эти ячейки будут переданы только на 4-ом цикле. Таким образом, попадание в очередь на входе ячейки блокирует передачу последующих ячеек даже если выходные каналы для их передачи свободны. Чтобы исключить блокировку такого рода можно организовать очередь не на входе, а на выходе коммутатора. При этом для коммутатора с 1024 входами теоретически может понадобиться 1024 буфера на каждом выходе. Реально число таких буферов значительно меньше. Такая схема ATM-коммутатора (8*8) показана на рис. 4.3.5.21.

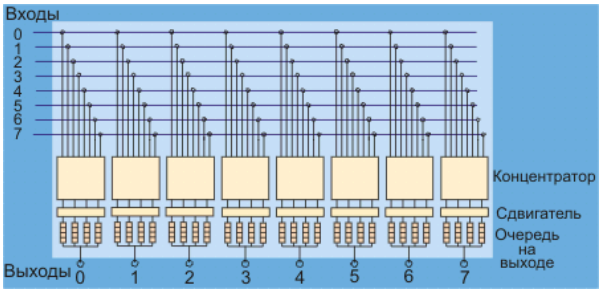


Рис. 4.3.5.21. Схема переключателя с организацией очередей на выходе

Концентратор выбирает N ячеек для помещения в очередь (предполагается, что максимальная длина очереди может быть равна N). Выходной буфер уже заполнен, ячека может быть потеряна. При построении АТМ-коммутаторов часто используется схема сети с многокаскадными соединениями.

Было время, когда сети ATM обладали наивысшим быстродействием, а это стримулировало адаптацию этого протокола для локальных сетейю И такая модификация (LANE) была разработана. LANE осуществляет взаимодействие соединенных посредством ATM оконечных станций для сегментов LAN. LANE не оказывает воздействия на работу сети ATM и не требует никаких специальных модификаций протокола. После внедрения FE и тем более GE область использования техники LANE сместилась в область WAN. LANE работает как система клиент-сервер, ее основная задача установление соответствия между адресами MAC и ATM.

9. Маршрутизация: маршрутизация первого уровня.

См. 7 вопрос. А лучше лекции.

10. Протокол SMTP. Модель, основные команды, безопасность, производительность.

Сообщения форматированы по правилам виртуального сетевого терминала (NVT), то есть в NVT ASCII. NVT подобен виртуальному сетевому протоколу и нужен затем, чтобы скрыть различия в восприятии разными компьютерами разных символов, например переводов каретки, переводов строки, маркеров конца строки, очистки экрана и т. д. Символ в NVT состоит из семи битов набора ASCII и является буквой, цифрой или знаком пунктуации. Семи битный набор ASCII часто называется NVT ASCII.

Модель протокола.

Взаимодействие в рамках SMTP строится по принципу двусторонней связи, которая устанавливается между отправителем и получателем почтового сообщения. При этом отправитель инициирует соединение и посылает запросы на обслуживание, а получатель - отвечает на эти запросы. Фактически отправитель выступает в роли клиента, а получатель - сервера.

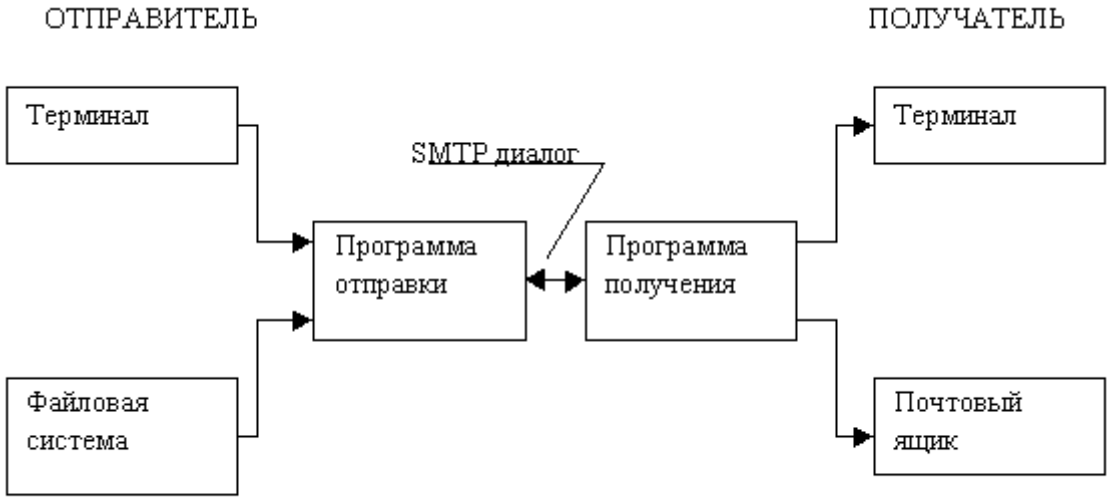


Рис. 5. *Схема взаимодействия по протоколу SMTP*

Канал связи устанавливается непосредственно между отправителем и получателем сообщения. При таком взаимодействии почта достигает абонента в течение нескольких секунд после отправки.

Команды SMTP

Простой протокол передачи почты обеспечивает двухсторонний обмен сообщениями между локальным клиентом и удаленным сервером MTA. MTA-клиент шлет команды MTA-серверу, а он, в свою очередь, отвечает клиенту. Другими словами, протокол SMTP требует получать ответы (они описаны в этой главе) от приемника команд SMTP. Обмен командами и ответами на них называется почтовой транзакцией (mail transaction). Данные, как мы уже говорили, передаются в формате NVT ASCII. Кроме того, команды тоже передаются в формате NVT ASCII. Команды передаются в форме ключевых слов, а не специальных символов, и указывают на необходимость совершить ту или иную операцию. В табл.1 приведен список ключевых слов (команд), определенный в спецификации SMTP - RFC 821.

Таблица 1. Команды простого протокола передачи почты (SMTP)

Команда	Обязательна	Описание
HELO	X	Идентифицирует модуль-передатчик для модуля-приемника (hello).
MAIL	X	Начинает почтовую транзакцию, которая завершается передачей

		данных в один или несколько почтовых ящиков (mail).
RCPT	X	Идентифицирует получателя почтового сообщения (recipient).
DATA		Строки, следующие за этой командой, рассматриваются получателем как данные почтового сообщения. В случае SMTP, почтовое сообщение заканчивается комбинацией символов: CRLF-точка-CRLF.
RSET		Прерывает текущую почтовую транзакцию (reset).
NOOP		Требует от получателя не предпринимать никаких действий, а только выдать ответ ОК. Используется главным образом для тестирования.(No operation).
QUIT		Требует выдать ответ ОК и закрыть текущее соединение.
VRFY		Требует от приемника подтвердить, что ее аргумент является действительным именем пользователя. (См. примечание.).
SEND		Начинает почтовую транзакцию, доставляющую данные на один или несколько терминалов (а не в почтовый ящик).
SOML		Начинает транзакцию MAIL или SEND, доставляющую данные на один или несколько терминалов или в почтовые ящики.
SAML		Начинает транзакцию MAIL и SEND, доставляющие данные на один или несколько терминалов и в почтовые ящики.
EXPN		Команда SMTP-приемнику подтвердить, действительно ли аргумент является адресом почтовой рассылки и если да, вернуть адрес получателя сообщения (expand).
HELP		Команда SMTP-приемнику вернуть сообщение-справку о его командах.
TURN		Команда SMTP-приемнику либо сказать ОК и поменяться ролями, то есть стать STMP- передатчиком, либо послать сообщение-отказ и остаться в роли SMTP-приемника.

Примечание: В RFC 821 сказано, что команда VRFY не является обязательной для минимального набора команд SMTP. Однако в RFC 1123 <Требования для сетевых компьютеров Internet - приложения и обеспечение работы> (Requirements for Internet Hosts - Application and Support,Braden, 1989), команда VRFY фигурирует в списке обязательных для Internet команд реализации SMTP.

В соответствии со спецификацией команды, помеченные крестиком (X) в табл.1, обязаны присутствовать в любой реализации SMTP. Остальные команды SMTP могут быть реализованы дополнительно. Каждая SMTP-команда должна заканчиваться либо пробелом (если у нее есть аргумент), либо комбинацией CRLF. В описании команд употреблялось слово <данные", а не <сообщение>. Этим подчеркивалось, что, кроме текста, SMTP позволяет передавать и двоичную информацию, например графические или звуковые файлы. Другими словами, SMTP способен передавать данные любого содержания, а не только текстовые сообщения. Это значит, что, рассматривая вопросы, касающиеся SMTP, не забывайте, что термин "сообщениее" обозначает не только текстовые данные.

Последовательность команд SMTP

Как мы уже отмечали, SMTP обеспечивает двухстороннюю связь между агентами передачи почты (MTA), клиентом и сервером. Клиенты шлют команды серверу, а серверы отвечают клиентам. Однако SMTP оговаривает последовательность SMTP-команд. Лучший способ понять это - взглянуть на образец почтовой транзакции. Следующий пример (он взят целиком из RFC 821) демонстрирует типичную почтовую транзакцию. В примере фигурирует мистер Smith (на компьютере usc.edu), посылающий сообщения мистерам Jones, Green и

Brown (на компьютере mit.edu). Агент передачи почты хоста mit.edu принимает почту для мистеров Jones и Brown, однако не знает, где расположен почтовый ящик мистера Green.

Для целей дальнейшего повествования каждой строке присвоен номер и обозначено, кому они принадлежат - передатчику или приемнику. Текст справа от слов <RECEIVER> или <SENDER> содержит действительно передаваемые данные. Трехзначные цифровые комбинации в начале передаваемых строк обозначают коды ответа (их значение объясняется позже). Ответ SMTP похож на сообщения-подтверждения о доставке, поскольку появляется лишь в том случае, когда приемник получил данные.

1	RECEIVER	220 mit.edu Simple Mail Transfer Service Ready
2	SENDER	HELO usc.edu
3	RECEIVER	250 mit.edu
4	SENDER	MAIL FROM: <Smith@usc.edu>
5	RECEIVER	250 OK
6	SENDER	RCPT TO:<Jones@mit.edu>
7	RECEIVER	250 OK
8	SENDER	RCPT TO:<Green@mit.edu>
9	RECEIVER	550 No such user here
10	SENDER	RCPT TO:<Brown@mit.edu>
11	RECEIVER	250 OK
12	SENDER	DATA
13	RECEIVER	354 Start mail input; end with <CRLF>.<CRLF>
14	SENDER	Blah blah blah...
15	SENDER	...etc. etc. etc.
16	SENDER	.
17	RECEIVER	250 OK
18	SENDER	QUIT
19	RECEIVER	221 mit.edu Service closing transmission channel

Как видно из строки 1, когда SMTP-клиент устанавливает TCP-соединение с портом протокола 25, SMTP-сервер отвечает кодом 220. Это означает, что соединение успешно установлено:

1. RECEIVER: 220 mit.edu Simple Mail Transfer Service Ready

После того как MTA компьютеров mit.edu и usc.edu установили соединение и обменялись приветствием, первой командой, согласно спецификации, должна быть команда HELO. Как указано в строке 2, SMTP-клиент передает HELO, указывая имя своего компьютера в качестве аргумента. Другими словами, он сообщает: <Привет, я - usc.edu>. Команда HELO употребляется с аргументом, как показано ниже:

2. SENDER: HELO usc.edu

В ответ на HELO приемник выдает код 250, сообщая передатчику о том, что команда принята и обработана:

3. RECEIVER: 250 mit.edu

После установления TCP-соединения и идентификации (при помощи HELO) SMTP-клиент приступает к почтовой транзакции. Для начала он выполняет одну из следующих команд: MAIL, SEND, SOML или SAML. В нашем примере использована команда MAIL:

4. SENDER: MAIL FROM:<Smith@usc.edu>

Все четыре команды, MAIL, SEND, SOML и SAML, имеют одинаковый синтаксис:
MAIL <пробел> FROM:<reverse-path> <carriage-return line-feed>

Примечание: Команды SEND, SOML и SAML дополнительны и используются довольно редко.

Аргумент <обратный путь> (reverse path) указывает серверу, кому в случае ошибки отослать соответствующее сообщение. Мы еще рассмотрим его подробнее. На данный момент для нас важно, что в аргументе содержится адрес источника сообщения (в нашем случае, Smith@usc,edu). После того как сервер выдал код ответа 250 (строка 5), согласившись обработать сообщение от Smith@usc.edUt необходимо указать получателя сообщения. Это делается при помощи команды RCPT. Команда RCPT имеет аргумент - имя получателя. На одну команду приходится только одно имя, поэтому, если получателей несколько, команда RCPT выдается несколько раз. В нашем примере команды RCPT выполняются в строках 6, 8 и 10. Синтаксис RCPT похож на синтаксис команды MAIL:
RCPT <пробел> TO:<forward-path> <CRLF>

Однако, в отличие от MAIL, аргумент RCPT начинается со слова <TO:>. Содержимое аргумента - путь передачи сообщения (forward path), а не обратный путь. На данный момент для нас важно, что в пути передачи сообщения указано имя почтового ящика получателя. Выдав команду RCPT, MTA-клиент ожидает получить ответ с кодом 250. Однако в ответ на восьмую строку

8. SENDER: RCPT TO:<Green@mit.edu>

сервер отвечает кодом 550:

9. RECEIVER: 550 No such user here

Код ответа 550 означает, что MTA не в состоянии выполнить запрос клиента, поскольку не знает, как доставить почту указанному пользователю. То есть, скорее всего, у мистера по фамилии Green нет почтового ящика (Green@mit,edu) на этом компьютере. В протоколе SMTP сказано, что сервер обязан информировать клиента об отсутствии почтового ящика получателя сообщения. Однако в спецификации SMTP ничего не говорится о том, как клиент должен реагировать на это сообщение.

После того как посланы все команды RCPT, клиент начинает передачу данных при помощи команды DATA. В строке 12 показано, как MTA-клиент (передатчик) высылает команду DATA, в строке 13 - как сервер отвечает кодом 354. Этот код означает, что передача данных разрешена и должна заканчиваться комбинацией CRLF-<точка>-CRLF (новой строкой, содержащей только точку).

12. SENDER: DATA
13. RECEIVER: 354 Start mail input; end with <CRLF>.<CRLF>

После того как получен код 354, клиент может начать передачу данных. MTA-сервер, в свою очередь, помещает принятые данные в очереди входящих сообщений. Сервер не высылает никаких ответов до тех пор, пока не получит комбинацию CRLF-точка-CRLF от клиента,

означающую конец передачи данных. Как показано в строках 16 и 17, в ответ на полученную комбинацию CRLF-<точка>-CRLF, сервер выдает код 250. Как мы уже говорили, код ответа 250 означает успешное окончание операции:
16. SENDER: .
17. RECEIVER: 250 OK

Для того чтобы закончить почтовую транзакцию, клиент, по правилам SMTP, обязан послать команду QUIT. Сервер, в свою очередь, отвечает кодом 221. Этот код подтверждает клиенту, что соединение будет закрыто, после чего соединение действительно закрывается:

18. SENDER: QUIT
19. RECEIVER: 221 mit.edu Service closing transmission channel

В любой момент во время транзакции клиент может использовать команды NOOP, HELP, EXPN и VRFY. В ответ на каждую команду сервер высылает клиенту определенную информацию. Конечно, в зависимости от ответа клиент может предпринять определенные действия, однако спецификация SMTP ничего не говорит по этому поводу. Например, клиент-MTA может передать команду VRFY для того, чтобы убедиться, что имя пользователя действительно. Если сервер ответит, что данного имени не существует, клиент MTA может не передавать почту для этого пользователя. В спецификации SMTP, однако, на этот счет нет никаких указаний - клиент может ничего не делать в ответ на команду VRFY. MTA-клиент может ничего не делать также в ответ на команды NOOP, HELP и EXPN - ответственность целиком лежит на разработчике конкретной реализации MTA.

Коды ответов SMTP

В спецификации SMTP требуется, чтобы сервер отвечал на каждую команду SMTP-клиента. MTA-сервер отвечает трехзначной комбинацией цифр, называемой кодом ответа. Вместе с кодом ответа, как правило, передается одна или несколько строк текстовой информации.

Примечание: Несколько строк текста, как правило, сопровождают только команды EXPN и HELP.В спецификации SMTP, однако, ответ на любую команду может состоять из нескольких строк текста.

Каждая цифра в коде ответа имеет определенный смысл. Первая цифра означает, было ли выполнение команды успешно (2), неуспешно (5) или еще не закончилось (3). Как указано в приложении E документа RFC 821, простой SMTP-клиент может анализировать только первую цифру в ответе сервера, и на основании ее продолжать свои действия. Вторая и третья цифры кода ответа разъясняют значение первой. Если вы разрабатываете SMTP-приложение, обязательно изучите конструкцию всех кодов SMTP-ответа. То, как коды составлены в самом SMTP - превосходный образец грамотного подхода к делу. В табл.2 приведены возможные значения кодов ответа SMTP, определенные в RFC 821.

Таблица 2. Коды ответа SMTP и их значение
Код

Значение
211 Ответ о состоянии системы или помощь
214 Сообщение-подсказка (помощь)
220 <имя_домена> служба готова к работе
221 <имя_домена> служба закрывает канал связи
250 Запрошенное действие почтовой транзакции успешно завершилось

251 Данный адресат не является местным; сообщение будет передано по маршруту <forward-path>
354 Начинай передачу сообщения. Сообщение заканчивается комбинацией CRLF-точка-CRLF
421 <имя_домена> служба недоступна; соединение закрывается
450 Запрошенная команда почтовой транзакции не выполнена, так как почтовый ящик недоступен
451 Запрошенная команда не выполнена; произошла локальная ошибка при обработке сообщения
452 Запрошенная команда не выполнена; системе не хватило ресурсов
500 Синтаксическая ошибка в тексте команды; команда не опознана
501 Синтаксическая ошибка в аргументах или параметрах команды
502 Данная команда не реализована
503 Неверная последовательность команд
504 У данной команды не может быть аргументов
550 Запрошенная команда не выполнена, так как почтовый ящик недоступен
551 Данный адресат не является местным; попробуйте передать сообщение по маршруту <forward-path>
552 Запрошенная команда почтовой транзакции прервана; дисковое пространство, доступное системе, переполнилось
553 Запрошенная команда не выполнена; указано недопустимое имя почтового ящика
554 Транзакция не выполнена

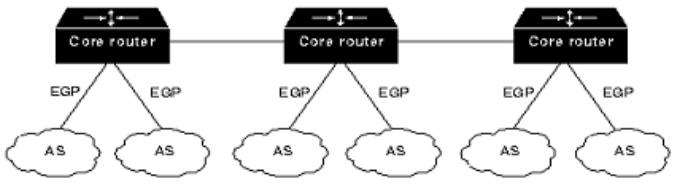
11. Протокол маршрутизации EGP

Протокол внешних маршрутизаторов (Exterior Gateway Protocol-EGP) является протоколом междоменной досягаемости, который применяется в Internet - международной сети, об'единяющей университеты, правительственные учреждения, научно-исследовательские организации и частные коммерческие концерны. EGP документально оформлен в Запросах для Комментария (RFC) 904, опубликованных в апреле 1984 г.

Являясь первым протоколом внешних маршрутизаторов, который получил широкое признание в Internet, EGP сыграл важную роль. К сожалению, недостатки EGP стали более очевидными после того, как Internet стала более крупной и совершенной сетью. Из-за этих недостатков EGP в настоящее время не отвечает всем требованиям Internet и заменяется другими протоколами внешних маршрутизаторов, такими, как Протокол граничных маршрутизаторов (Border Gateway Protocol - BGP) и Протокол междоменной маршрутизации (Inter-Domain Routing Protocol - IDRP).

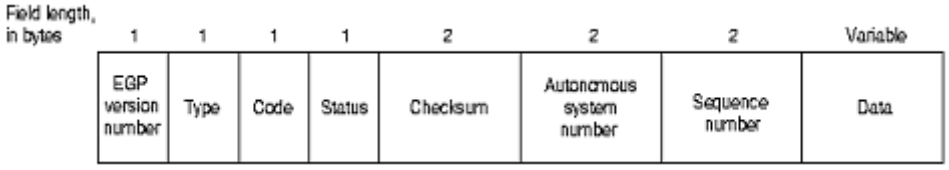
Основы технологии

EGP первоначально предназначался для передачи информации о досягаемости в стержневые маршрутизаторы ARPANET и получения ее от них. Информация передавалась из отдельных узлов источника, находящихся в различных административных доменах, называемых автономными системами (AS), вверх в стержневые маршрутизаторы, которые передавали эту информацию через стержневую область до тех пор, пока ее можно было передать вниз к сети пункта назначения, находящейся в пределах другой AS. Эти взаимоотношения между EGP и другими компонентами ARPANET показаны на Рисунке 4.29.



Несмотря на то, что EGP является динамическим протоколом маршрутизации, он использует очень простую схему. Он не использует показатели, и следовательно, не может принимать по настоящему интеллектуальных решений о маршрутизации. Корректировки маршрутизации EGP содержат информацию о досягаемости сетей. Другими словами, они указывают, что в определенные сети попадают через определенные маршрутизаторы.
EGP имеет три основных функции. Во-первых, маршрутизаторы, работающие с EGP, организуют для себя определенный набор соседей. Соседи - это просто другие маршрутизаторы, с которыми какой-нибудь маршрутизатор хочет коллективно пользоваться информацией о досягаемости сетей; какие-либо указания о географическом соседстве не включаются. Во-вторых, маршрутизаторы EGP опрашивают своих соседей для того, чтобы убедиться в их работоспособности. В-третьих, маршрутизаторы EGP отправляют сообщения о корректировках, содержащих информацию о досягаемости сетей в пределах своих AS.

Формат пакета EGP



Первым полем в заголовке пакета EGP является поле номера версии EGP (EGP version number). Это поле обозначает текущую версию EGP и проверяется приемными устройствами для определения соответствия между номерами версий отправителя и получателя.
Следующим полем является поле типа (type), которое обозначает тип сообщения. EGP выделяет 5 отдельных типов сообщения.
За полем типа следует поле кода (code). Это поле определяет различие между подтипами сообщений.
Следующее поле - поле состояния (status), которое содержит информацию о состоянии, зависящую от сообщения. В число кодов состояния входят коды недостатка ресурсов (insufficient resources), неисправных параметров (parameter problem), нарушений протокола (protocol violation), и другие.
За полем состояния идет поле контрольной суммы (checksum). Контрольная сумма используется для обнаружения возможных проблем, которые могли появиться в пакете в результате транспортировки.
За полем контрольной суммы идет поле номера автономной системы (autonomous system number). Оно обозначает AS, к которой принадлежит маршрутизатор- отправитель.
Последним полем заголовка пакета EGP является поле номера последовательности (sequence number). Это поле позволяет двум маршрутизаторам EGP, которые обмениваются сообщениями, согласовывать запросы с ответами. Когда определен какой-нибудь новый сосед, номер последовательности устанавливается в исходное нулевое значение и инкрементируется на единицу с каждой новой транзакцией запрос-ответ.
За заголовком EGP идут дополнительные поля. Содержимое этих полей различается в зависимости от типа сообщения (определяемого полем типа).

Типы сообщений

За заголовком EGP идут дополнительные поля. Содержимое этих полей различается в зависимости от типа сообщения (определяемого полем типа).

Приобретение соседа

Сообщение "приобретение соседа" включает в себя интервал приветствия (hello interval) и интервал опроса (poll interval). Поле интервала приветствия определяет период интервала проверки работоспособности соседей. Поле интервала опроса определяет частоту корректировки маршрутизации.

Досягаемость соседа

Сообщения о досягаемости соседа не имеют отдельных полей в числе полей, идущих за заголовком EGP. Эти сообщения используют поле кода для указания различия между при-

ветственным сообщением и ответом на приветственное сообщение. Выделение функции оценки досягаемости из функции корректировки маршрутизации уменьшает сетевой трафик, т.к. изменения о досягаемости сетей обычно появляются чаще, чем изменения параметров маршрутизации. Любой узел EGP заявляет об отказе одного из своих соседей только после того, как от него не был получен определенный процент сообщений о досягаемости.

Опрос

Чтобы обеспечить правильную маршрутизацию между AS, EGP должен знать об относительном местоположении отдаленных хостов. Сообщение опроса позволяет маршрутизаторам EGP получать информацию о досягаемости сетей, в которых находятся эти машины. Такие сообщения имеют только одно поле помимо обычного заголовка - поле сети источника IP (source network). Это поле определяет сеть, которая должна использоваться в качестве контрольной точки для запроса.

Корректиравка маршрутизации

Сообщения о корректировке маршрутизации дают маршрутизаторам EGP возможность указывать местоположение различных сетей в пределах своих AS. В дополнение к обычному заголовку эти сообщения включают несколько дополнительных полей. Поле числа внутренних маршрутизаторов (number of interior gateways) указывает на число внутренних маршрутизаторов, появляющихся в сообщении. Поле числа внешних маршрутизаторов (number of exterior gateways) указывает на число внешних маршрутизаторов, появляющихся в сообщении. Поле сети источника IP (IP source network) обеспечивает адрес IP той сети, от которой измерена досягаемость. За этим полем идет последовательность блоков маршрутизаторов (gateway blocks). Каждый блок маршрутизаторов обеспечивает адрес IP какого-нибудь маршрутизатора и перечень сетей, а также расстояний, связанных с достижением этих сетей.

В пределах одного блока маршрутизатора EGP перечисляет сети по расстояниям. Например, на расстоянии три может быть четыре сети. Эти сети перечислены по адресам. Следующей группой сетей могут быть сети, находящиеся на расстоянии 4, и т.д.

EGP не расшифровывает показатели расстояния, содержащиеся в сообщениях о корректировке маршрутов. EGP фактически использует поле расстояния для указания существования какого-либо маршрута; значение расстояния может быть использовано только для сравнения трактов, если эти тракты полностью находятся в пределах одного конкретного AS. По этой причине EGP является скорее протоколом досягаемости, чем протоколом маршрутизации. Это ограничение приводит также к ограничениям в структуре Internet. Характерно, что любая часть EGP сети Internet должна представлять собой структуру дерева, у которого стержневой маршрутизатор является корнем, и в пределах которого отсутствуют петли между другими AS. Это ограничение является основным ограничением EGP; оно стало причиной его постепенного вытеснения другими, более совершенными протоколами внешних маршрутизаторов.

Сообщения о неисправностях

Сообщения о неисправностях указывают на различные сбойные ситуации. В дополнение к общему заголовку EGP сообщения о неисправностях обеспечивают поле причины (reason), за которым следует заголовок сообщения о неисправности (message header). В число типичных неисправностей (причин) EGP входят неисправный формат заголовка EGP (bad EGP header format), неисправный формат поля данных EGP (bad EGP data field format), чрезмерная скорость опроса (excessive polling rate) и невозможность достижения информации (unavailability of reachability information). Заголовок сообщения о неисправности состоит из первых трех 32-битовых слов заголовка EGP.

12. Протокол маршрутизации RIP.

Этот протокол маршрутизации предназначен для сравнительно небольших и относительно однородных сетей (алгоритм Белмана-Форда). Маршрут здесь характеризуется вектором расстояния до места назначения. Предполагается, что каждый маршрутизатор является от-

правной точкой нескольких маршрутов до сетей, с которыми он связан. Описания этих маршрутов хранится в специальной таблице, называемой маршрутной. Таблица маршрутизации RIP содержит по записи на каждую обслуживаемую машину (на каждый маршрут). Запись должна включать в себя:
IP-адрес места назначения.
Метрика маршрута (от 1 до 15; число шагов до места назначения).
IP-адрес ближайшего маршрутизатора (gateway) по пути к месту назначения.
Таймеры маршрута.
Первым двум полям записи мы обязаны появлению термина вектор расстояния (место назначение - направление; метрика - модуль вектора). Периодически (раз в 30 сек) каждый маршрутизатор посылает широковещательно копию своей маршрутной таблицы всем соседям-маршрутизаторам, с которыми связан непосредственно. Маршрутизатор-получатель просматривает таблицу. Если в таблице присутствует новый путь или сообщение о более коротком маршруте, или произошли изменения длин пути, эти изменения фиксируются получателем в своей маршрутной таблице. Протокол RIP должен быть способен обрабатывать три типа ошибок:

1. Циклические маршруты. Так как в протоколе нет механизмов выявления замкнутых маршрутов, необходимо либо слепо верить партнерам, либо принимать меры для блокировки такой возможности.
2. Для подавления нестабильностей RIP должен использовать малое значение максимально возможного числа шагов (<16).
3. Медленное распространение маршрутной информации по сети создает проблемы при динамичном изменении маршрутной ситуации (система не поспевает за изменениями). Малое предельное значение метрики улучшает сходимость, но не устраняет проблему.

Несоответствие маршрутной таблицы реальной ситуации типично не только для RIP, но характерно для всех протоколов, базирующихся на векторе расстояния, где информационные сообщения актуализации несут в себе только пары кодов: адрес места назначение и расстояние до него. Пояснение проблемы дано на рис. 4.4.1.11.1 ниже.

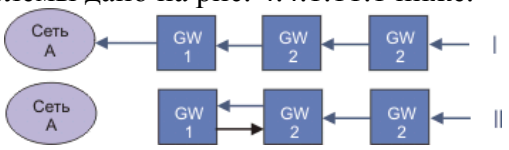


Рис. 4.4.11.1. Иллюстрация, поясняющее возникновение циклических маршрутов при использовании вектора расстояния.

На верхней части рисунка показана ситуация, когда маршрутизаторы указывают маршрут до сети в соответствии со стрелками. На нижней части связь на участке GW1 <Сеть a> оборвана, а GW2 по-прежнему продолжает оповещать о ее доступности с числом шагов, равным 2. При этом GW1, восприняв эту информацию (если GW2 успел передать свою маршрутную информацию раньше GW1), может перенаправить пакеты, адресованные сети A, на GW2, а в своей маршрутной таблице будет характеризовать путь до сети A метрикой 3. При этом формируется замкнутая петля маршрутов. Последующая широковещательная передача маршрутных данных GW1 и GW2 не решит эту проблему быстро. Так после очередного обмена путь от gw2 до сети A будет характеризоваться метрикой 5. Этот процесс будет продолжаться до тех пор, пока метрика не станет равной 16, а это займет слишком много циклов обмена маршрутной информацией.
Проблема может быть решена следующим образом. Маршрутизатор запоминает, через какой интерфейс получена маршрутная информация, и через этот интерфейс эту информацию уже не передает. В рассмотренном выше примере GW2 не станет посылать информацию о пути к сети A маршрутизатору GW1, от которого он получил эти данные. В этом случае в маршрутной таблице GW1 путь до A исчезнет сразу. Остальные маршрутизаторы узнают о недости-

жимости сети А через несколько циклов. Существуют и другие пути преодоления медленных переходных процессов. Если производится оповещение о коротком пути, все узлы-получатели воспринимают эти данные немедленно. Если же маршрутизатор закрывает какой-то путь, его отмена фиксируется остальными лишь по тайм-ауту. Универсальным методом исключения ошибок при маршрутизации является использование достаточно большой выдержки, перед тем как использовать информацию об изменении маршрутов. В этом случае к моменту изменения маршрута эта информация станет доступной всем участникам процесса маршрутизации. Но все перечисленные методы и некоторые другие известные алгоритмы, решая одну проблему, часто вносят другие. Многие из этих методов могут при определенных условиях вызвать лавину широковещательных сообщений, что также дезорганизует сеть. Именно малая скорость установления маршрутов в RIP (и других протоколах, ориентированных на вектор расстояния) и является причиной их постепенного вытеснения другими протоколами.

Но даже усовершенствование, изложенное выше, не всегда срабатывает. На рис. 4.4.11.1а приведен пример, когда переходной процесс, несмотря на усовершенствование будет идти долго. При обрыве связи В-Г узлы А и Б сообщают узлу В, что они потеряли связь с узлом Г. Узел В делает вывод, что Г не достижим, о чем и сообщает узлам А и Б. К сожалению А знает, что Б имеет проход к Г длиной 2, из чего он делает вывод о достижимости Г за три шага. Аналогично рассуждает Б о возможности достижимости Г через А. Далее при последующих рассылках метрика доступности Г, характеризуется все большими значениями, до тех пор пока не станет равной "бесконечности".

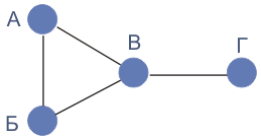


Рис. 4.4.11.1а. Пример топологии, где переходной процесс осуществляется медленно, даже при усовершенствовании алгоритма.

В RIP сообщения инкапсулируются в udp-дейтограммы, при этом передача осуществляется через порт 520. В качестве метрики RIP использует число шагов до цели. Если между отправителем и приемником расположено три маршрутизатора (gateway), считается, что между ними 4 шага. Такой вид метрики не учитывает различий в пропускной способности или загруженности отдельных сегментов сети. Применение вектора расстояния не может гарантировать оптимальность выбора маршрута, ведь, например, два шага по сегментам сети ethernet обеспечат большую пропускную способность, чем один шаг через последовательный канал на основе интерфейса RS-232.

Маршрут по умолчанию имеет адрес 0.0.0.0 (это верно и для других протоколов маршрутизации). Каждому маршруту ставится в соответствие таймер тайм-аута и "сборщика мусора". Тайм-аут-таймер сбрасывается каждый раз, когда маршрут инициализируется или корректируется. Если со времени последней коррекции прошло 3 минуты или получено сообщение о том, что вектор расстояния равен 16, маршрут считается закрытым. Но запись о нем не стирается, пока не истечет время "уборки мусора" (2мин). При появлении эквивалентного маршрута переключения на него не происходит, таким образом, блокируется возможность осцилляции между двумя или более равноценными маршрутами. Формат сообщения протокола RIP имеет вид, показанный на рис. 4.4.11.2. Поле команда определяет выбор согласно следующей таблице:

Таблица 4.4.11.1. Значения кодов поля команда

Команда	**Значение**
1	Запрос на получение частичной или полной маршрутной информации;
2	Отклик, содержащий информацию о расстояниях из маршрутной таблицы отпра-

	вителя;
3	Включение режима трассировки (устарело);
4	Выключение режима трассировки (устарело);
5-6	Зарезервированы для внутренних целей sun microsystem.

Поле версия для RIP равно 1 (для RIP-2 двум). Поле набор протоколов сети i определяет набор протоколов, которые используются в соответствующей сети (для Интернет это поле имеет значение 2). Поле расстояние до сети i содержит целое число шагов (от 1 до 15) до данной сети. В одном сообщении может присутствовать информация о 25 маршрутах. При реализации RIP можно выделить следующие режимы:

Инициализация, определение всех "живых" интерфейсов путем посылки запросов, получение таблиц маршрутизации от других маршрутизаторов. Часто используются широковещательные запросы.

Получен запрос. В зависимости от типа запроса высылается адресату полная таблица маршрутизации, или проводится индивидуальная обработка.

Получен отклик. Проводится коррекция таблицы маршрутизации (удаление, исправление, добавление).

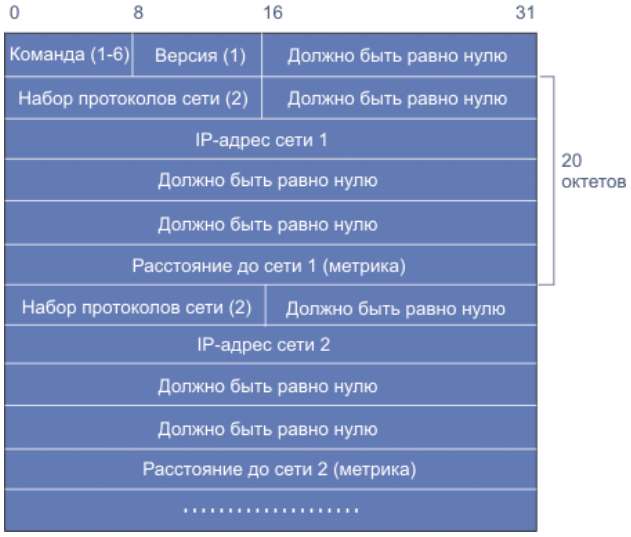


Рис. 4.4.11.2. Формат сообщения RIP.

Регулярные коррекции. Каждые 30 секунд вся или часть таблицы маршрутизации посылается всем соседним маршрутизаторам. Могут посылаться и специальные запросы при локальном изменении таблицы. RIP достаточно простой протокол, но, к сожалению не лишенный недостатков:

1. RIP не работает с адресами субсетей. Если нормальный 16-бит идентификатор ЭВМ класса B не равен 0, RIP не может определить является ли не нулевая часть субсетевым ID, или полным IP-адресом.
2. RIP требует много времени для восстановления связи после сбоя в маршрутизаторе (минуты). В процессе установления режима возможны циклы.
3. Число шагов важный, но не единственный параметр маршрута, да и 15 шагов не предел для современных сетей.

Протокол RIP-2 (RFC-1388, 1993 год) является новой версией RIP, которая в дополнение к широковещательному режиму поддерживает мультикастинг; позволяет работать с масками субсетей. На рис. 4.4.11.3 представлен формат сообщения для протокола RIP-2. Поле маршрутный демон является идентификатором резидентной программы-маршрутизатора. Поле метка маршрута используется для поддержки внешних протоколов маршрутизации, сюда записываются коды автономных систем. При необходимости управления доступом можно

использовать первые 20 байт с кодом набора протоколов сети 0xFFFF и меткой маршрута =2. Тогда в остальные 16 байт можно записать пароль.

Рис. 4.4.11.3 Формат сообщений протокола RIP-2

Проблемы аутентификации в протоколе RIP-2 с использованием дайджестов MD5 обсуждаются в документе RFC-2082

13. Сети X.25.

В 1976 году был принят стандарт X.25, который стал основой всемирной системы PSPDN (Packet-Switched Public Data Networks), базирующейся на 7-уровневой модели ISO OSI. Стандарт X.25 был усовершенствован в 1984. X.25 - протокол (ISO 8208:1989; RFC-887, -1381, -1382, -1461, -1598, -1613), который определяет синхронный интерфейс между терминальным оборудованием (DTE - Data Terminal Equipment) и оборудованием передачи данных (DCE - Data Communication Equipment) для терминалов, работающих в пакетном режиме. По существу это протокол связи оборудования с сетью. Главный недостаток протокола X.25 - большие задержки отклика (типовое значение 0.6 сек). Терминалом может служить ЭВМ или любая другая система, удовлетворяющая требованиям X.25. Соединение DTE - DTE осуществляется через DCE. В протоколе X.25 DCE и DTE используют статистическое мультиплексирование с делением по времени. Одновременно могут реализовываться несколько обменных процессов. Схема взаимодействия DTE и DCE выглядит как:

DTE - <логический канал> - DCE <виртуальное соединение> - DCE - <логический канал> - DTE

Асинхронный старт-стопный терминал подключается к сети коммутации пакетов через пакетный адаптер данных ПАД (PAD - packet assemble/disassemble) и отвечает рекомендациям X.3, X.28 и X.29. Один ПАД обеспечивает интерфейс для 8, 16 или 24 асинхронных терминалов. Пакет данных состоит обычно из 128 байтов, которые передаются по адресу, содержащемуся в пакете. Но длина пакета может лежать в пределах 64-4096 байтов. Размер пакета также как и величина окна (число пакетов, принимаемых без подтверждения) определяются на фазе установления канала. Прежде чем пакет будет передан, необходимо установить связь между исходными ЭВМ/ПАД и адресуемыми ЭВМ/ПАД. Существуют два вида соединений: коммутируемый виртуальный канал (SVC) и постоянный виртуальный канал (PVC). Предусмотрены две процедуры доступа к каналу:

- Процедура доступа к каналу (LAP - link access procedure), в основе которой лежат симметричные операции режима асинхронного ответа (ARM - asynchronous response mode) протокола HDLC.
- Балансная процедура доступа к каналу (LAPB - link access procedure balanced) на основе асинхронного балансного режима (ABM - asynchronous balanced mode) протокола HDLC. Сетевой уровень реализуется с использованием 14 различных типов пакетов.

Виртуальный канал описывается в общем формате пакета, как "логический канал". Логический канал имеет идентификатор, состоящий из 12 бит. Этот идентификатор обычно состоит из номера группы (4 бита) и номера логического канала (8 бит). В группе может быть до 256 логических каналов (за исключением группы 0, которая может иметь только 255 логических каналов). Возможное число групп - 16, поэтому теоретически возможное число виртуальных каналов для каждого соединения x.25 равно 4095 (16x256-1).

Постоянный виртуальный канал (PVC - permanent virtual circuit) является аналогом выделенного канала.

Коммутируемый виртуальный канал (SVC - switched virtual circuit - напоминает традиционный телефонный вызов) реализует обмен данными. Имеются три типа коммутируемых виртуальных каналов, работающие в дуплексном режиме, но отличающиеся направлением устанавливаемых соединений: входящий SVC, двунаправленный SVC и выходящий SVC. Адресат каждого пакета распознается с помощью идентификатора логического канала (LCI) или номера логического канала (LCN).

SVC используются только на время соединения и становятся доступными для повторного использования после разъединения. Все типы пакетов, за исключением пакетов запроса повторного пуска, содержат идентификатор логического канала. Пакет запрос соединения в SVC является единственным типом пакетов, которые содержат адреса в соответствии с рекомендацией X.121.

Для установки выходящего соединения через svc ЭВМ выбирает логический канал с наибольшим номером в группе и посылает пакет запрос соединения, содержащий выбранный номер группы канала, адрес получателя (в соответствии с рекомендацией X.121) и в отдельных случаях свой собственный адрес. При установлении входящего соединения центр коммутации пакетов (ЦКП) выбирает свободный логический канал с наименьшим номером в группе каналов порта адресуемой ЭВМ и помещает этот логический номер группы и канала в пакет входящий запрос соединения. После того как соединение через svc установлено, ЭВМ направляют свои пакеты, используя номера своих логических групп/каналов, а ЦПК в сети осуществляет транспортировку пакетов и преобразование номеров логических групп/каналов. Как только установленное по svc логическое соединение разъединяется, номера логических групп/каналов на обоих концах соединения освобождаются и становятся доступными для повторного использования. Соответствие между ЦКП/портом, выделенным для терминального оборудования, адресами (согласно рекомендациям x.121) и номерами логических каналов известно в сети только ЦКП.

Выбор ЭВМ свободного канала с наибольшим номером при каждом выходящем соединении и выбор в ЦКП свободного канала с наименьшим номером для каждого входящего позволяют избежать конфликтов. С этой же целью используются две логические группы: одна только для входящих соединений, а другая только для выходящих. Перед подключением к сети пользователь должен определить, сколько pvc и svc требуется на каждую точку физического интерфейса x.25. Асинхронные терминалы подключаются к сети коммутации пакетов через встроенные или удаленные пакетные адаптеры данных (ПАД).

Встроенный ПАД обычно располагается вместе с ЦКП в его стойке. В этом случае каждый асинхронный терминал, расположенный в удаленном месте, подключается к своему встроенному ПАД через отдельный канал связи (протокол X.28). В альтернативном случае удаленный ПАД (небольшое отдельное устройство) может быть расположен в удаленном месте и подключается к своему ЦКП через канал связи (X.25). С помощью удаленного ПАД к ЦКП подключается 8-16 асинхронных терминалов.

Встроенный ПАД может быть совместно использован несколькими терминалами, расположенными в различных местах, в то время как удаленный ПАД обслуживает терминалы, расположенные обычно в одном месте. Существует еще один аспект размещения ПАД, связанный с помехами в каналах связи и использованием протоколов. Удаленный ПАД подключается к ЦКП на канальном уровне в соответствии с рекомендацией X.25. В качестве

протокола канала данных в рекомендации X.25 реализуется подмножество HDLC, обеспечивающее автоматическую повторную передачу данных в случае их искажения при возникновении помех в линии. Асинхронный терминал использует для диалога с групповым ПАД процедуры, описанные в рекомендации X.28, в которых не предусмотрена возможность повторной передачи в случае ошибки. Поэтому канал между синхронным терминалом и групповым ПАД не защищен от возникновения ошибок данных в результате линейных помех. Процедуры ПАД определены в рекомендациях МККТТ (см. приложение 10.1).

- Рекомендация X.3: "Пакетный адаптер данных (ПАД) в сети передачи данных общего пользования".
- Рекомендация X.28: "Интерфейс между терминальным оборудованием и оборудованием передачи данных (DCE) для старт-стопного оконечного оборудования, осуществляющего доступ к пакетному адаптеру данных в сетях общего пользования".
- Рекомендация X.29: "Процедуры обмена управляющей информацией между терминальным оборудованием пакетного типа и пакетным адаптером (ПАД)".

Основные функции ПАД соответствуют рекомендациям X.3:

- сборка символов (полученных от асинхронных терминалов) в пакеты;
- разборка полей данных в пакетах и вывод данных на асинхронные терминалы;
- управление процедурами установления виртуального соединения и разъединения, сброса и прерывания;
- обеспечение механизма продвижения пакетов при наличии соответствующих условий, таких как заполнение пакета, получение символа (сигнала) на передачу пакета, истечение времени ожидания;
- передача символов, включающих стартстопные сигналы и биты проверки на четность, по требованию подключенного асинхронного терминала;
- обнаружение сигнала разрыв соединения от асинхронного терминала;
- редактирование последовательностей команд ПАД.

В постоянном запоминающем устройстве ПАД хранятся параметры. Эти параметры могут быть установлены либо асинхронным терминалом, подключенным к ПАД, либо любой ЭВМ в сети, которая удовлетворяет условиям рекомендации X.29. В рекомендации X.29 МККТТ эти параметры названы управляющей информацией. Поэтому необходимо квалифицировать данные, проходящие между ЭВМ и ПАД, либо как управляющую информацию (сообщения ПАД), либо как собственно данные от асинхронного терминала.

Сеть X.25 предоставляет пользователю старт-стопного терминала средства, позволяющие выбрать параметры ПАД с заранее определенными значениями. Пользователь посылает в ПАД команду выбора профайла, которая включает идентификатор профайла. Этим определяется один из нескольких стандартных профайлов, хранящихся в ПАД.

Идентификатор профайла и параметр 11 ПАД (скорость терминала) включаются в "поле данных пользователя" пакетов типа запрос соединения, посылаемых ПАД. ЭВМ (ПАД) использует это поле, извлекая из него информацию о терминале, пославшем запрос.

Пакетный терминал является интеллектуальным устройством (например, ЭВМ, или внешним ПАД’ом), которое обеспечивает синхронный обмен с сетью на скорости 2400, 4800, 9600 бит/с или 48 Кбит/с, используя трехуровневый протокол X.25. Возможная схема подключения терминальных устройств к сети X.25 показана на рис. 4.3.2.1.

Из рисунка 4.3.2.1 видно, что подключение ЭВМ и другого терминального оборудования возможно как к встроенному, так и удаленному ПАД (протокол X.28), а также непосредственно к ЦКП (протокол X.25, X.29). Связи с удаленными объектами осуществляются через соответствующие модемы (на рисунке не показаны).

Для международного соединения необходимо указать код страны из трех цифр, а также набрать одну цифру 9 перед сетевым адресом пользователя. Таким образом, всего требуется не более 15 цифровых символов. Для установления коммутируемого соединения оператор

вначале вручную набирает номер ПАД и ждет подтверждения соединения с телефонным узлом общего пользования. Как только соединение установлено, оператор набирает 12-символьный код "сетевого идентификатора пользователя". ПАД обеспечивает операцию эхо-контроля, которая позволяет оператору терминала визуально проверять данные, посылаемые в ПАД. Наиболее серьезным недостатком встроенного ПАД является отсутствие какого-либо линейного протокола, предусматривающего устранение ошибок в данных, посылаемых от ПАД к терминалу. В удаленном ПАД предусмотрена процедура восстановления ошибочных данных, однако он подключается к сети как "пакетный терминал".

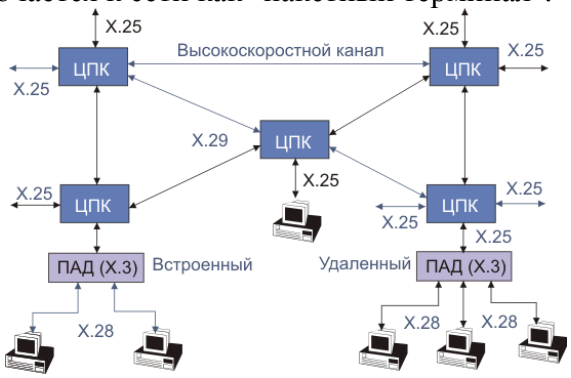


Рис. 4.3.2.1. Возможная топология сети X.25

Сетевой адрес пользователя состоит из 12 десятичных цифр. 1-4 - идентификатор сети передачи данных (3 - страна, 4 - сеть); 5-12 - национальный номер (5-7 местная область, 8-12 - местный номер). Международная система адресации для систем передачи данных общего пользователя описана в рекомендациях X.121 международного комитета по телефонии и телеграфии. Каждое подключение к сети коммутации пакетов имеет свой национальный номер. Протокол X.25 не определяет технику маршрутизации пакетов по сети. Для целей управления в сетях X.25 используется протокол snmp и база данных MIB (как и в сетях Интернет). Три базовых уровня протокола X.25 и схема потоков информации отображены на рис. 4.3.2.2.

Для подключения по виртуальному каналу ЭВМ/ПАД посылается пакет (call request), содержащий сетевой адрес пользователя. После подтверждения соединения и передачи/приема данных виртуальное соединение может быть разорвано путем передачи пакета (clear request), инициатором в этом случае выступает удаленная ЭВМ. При невозможности установить связь clear request посылается сетью. Такой пакет содержит два информационных октета. Первый содержит код причины, второй является диагностическим кодом. Ниже в таблице 4.3.2.1 приведены коды причин ошибки.

Таблица 4.3.2.1. Коды причины ошибки

Код причины	**Причина**
0x0	Удаленный сброс (remote cleared)
0x1	Адресат занят (number busy)
0x3	Нелегальный запрос (invalid facility request)
0x5	Перегрузка сети (network congestion)
0x9	Нарушен порядок (out of order)
0x11	Ошибка при выполнении удаленной процедуры
0xb	Доступ блокирован (access barred)
0xd	Не доступно, нет в наличии (not obtainable)

0x21	Несовместимость у адресата (ошибка при выполнении удаленной процедуры)
0x23	Ошибка при выполнении местной процедуры
0x29	Сигнал быстрой выборки не воспринят (fast select not accepted)

Один физический канал связи X.25 может поддерживать несколько коммутируемых виртуальных каналов. Постоянный виртуальный канал подобен выделенной линии - обмен возможен в любой момент. X.25 определяет первые три уровня соединения открытых систем (см. рис. 4.3.2.2).

1. физический x.21 (X.21bis)
2. канальный (HDLC - high data link communication - протокол высокого уровня управления каналом). Этот уровень и последующие реализуются программным образом.
3. сетевой (пакетный)

X.21 - универсальный интерфейс между оконечным оборудованием (DTE) и аппаратурой передачи данных (DCE) для синхронного режима работы в сетях общего пользования. X.21bis - тоже, но для модемов, удовлетворяющих рекомендациям серии V. Для канального уровня используется подмножество протокола HDLC (являющегося развитием стандарта SDLC IBM), обеспечивающее возможность автоматической повторной передачи в случае возникновения ошибок в линии.

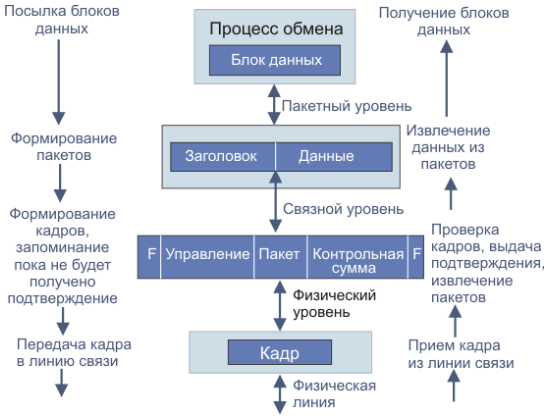


Рис. 4.3.2.2. Три уровня X.25

Формат кадра для протокола HDLC показан на рис. 4.3.2.3 (байты передаются, начиная с младшего бита):

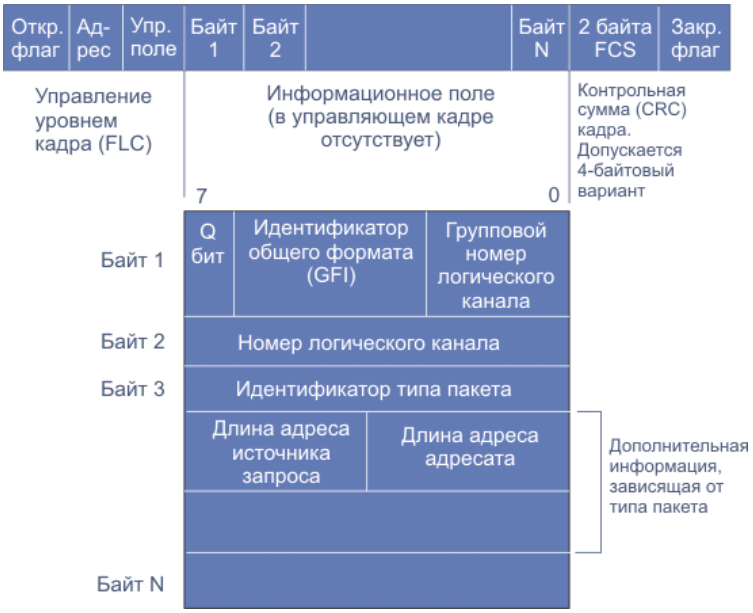


Рис. 4.3.2.3. Формат кадра X.25

Открывающий и закрывающий флаги для бит-ориентированного формата несут в себе код 0x7e. Когда не передается никакой информации, по каналу пересылается непрерывный поток флагов 01111110. Посылка более 6 единиц подряд воспринимается как флаг абортирования связи. Если необходимо передать информационную последовательность 01111110, после первых пяти единиц вводится дополнительный нуль, приемник восстанавливает истинную информацию, удаляя эти лишние нули. В случае байт-ориентированных кадров открывающий и завершающий флаги имеют по два байта [DLE (Символы кодов стандарта ISO 646-1973 (МТК-5, ГОСТ 13059-74). Здесь и далее используется русская терминология в соответствии со стандартом ГОСТ 26556-85, STX и DLE, ETX, соответственно, для информационного кадра и DLE, STX и DLE, ETX для управляющего]. Адрес в пакете X.25 занимает всего один байт, что определяет предельное число терминальных устройств, подключаемых к одному каналу. Кадр на уровне 2 имеет двухбайтовый заголовок, содержащий байт адреса и байт типа. Для нумерации кадров на уровне 2 используется 3 бита. При работе со скользящим окном откликов это позволяет иметь до 7 кадров в очереди. При использовании спутниковых каналов с большими задержками можно переходить в режим расширенной нумерации (7 бит), где длина очереди может достигать 128. Если удаленный партнер не способен работать в режиме расширенной нумерации, он отклонит запрос соединения. При работе в режиме расширенной нумерации возможно применение 3-байтовых заголовков вместо двухбайтовых.

Значения поля идентификатора общего формата (GFI - general format identifier) приведено в таблице 4.3.2.2. Бит 8 этого поля (Q) используется в информационных пакетах как индикатор уровня передаваемых данных. Групповой номер логического канала и номер логического канала присваиваются по соглашению с администрацией сети во время постановки на обслуживание. Поля групповой номер логического канала и номер логического канала присутствуют во всех пакетах кроме пакетов регистрации и повторного пуска, где они принимают нулевое значение.

Таблица 4.3.2.2. Значения кодов идентификатора общего формата (GFI)

Тип пакета	Модуль нумерации	Номера битов			
		8	7	6	5
Установка соединения	8 128	0 0	x x	0 1	1 0
Разрыв соединения, управление потоком, повторный пуск, регистрация, диагностика	8 128	0 0	0 0	0 1	1 0
Данные	8 128	x x	x x	0 1	1 0
Расширение	-	0	0	1	1

x - бит может принимать значения 0 или 1.

Допустимые значения кодов в поле тип пакета приведены в таблице 4.3.2.3.

Таблица 4.3.2.3. Значения кодов тип пакета

Тип пакета	Октет 3
Биты	8 7 6 5 4 3 2 1
Запрос	0 0 0 0 1 0 1 1
Запрос принят	0 0 0 0 1 1 1 1

Запрос завершения	0 0 0 1 0 0 1 1
Подтверждение завершения	0 0 0 1 0 1 1 1
Данные	x x x x x x x 0
Прерывание	0 0 1 0 0 0 1 1
Подтверждение прерывания	0 0 1 0 0 1 1 1
Готовность к приему по модулю 8 (RR)	x x x 0 0 0 0 1
Готовность к приему по модулю 128 (RR)	0 0 0 0 0 0 0 1
Неготовность к приему по модулю 8 (RNR)	x x x 0 0 1 0 1
Неготовность к приему по модулю 128 (RNR)	0 0 0 0 0 1 0 1
Запрос повторной установки	0 0 0 1 1 0 1 1
Подтверждение повторной установки	0 0 0 1 1 1 1 1
Запрос повторного пуска	1 1 1 1 1 0 1 1
Подтверждение повторного пуска	1 1 1 1 1 1 1 1
Диагностика	1 1 1 1 0 0 0 1
Запрос регистрации	1 1 1 1 0 0 1 1
Подтверждение регистрации	1 1 1 1 0 1 1 1

x - отмечет разряды, которые могут принимать значения 0 или 1.

Четырехбитовые поля длина адреса отправителя и длина адреса получателя характеризуют длины последующих полей переменной длины. Длина выражается в полуоктетах. Далее следуют соответствующие адреса. В каждом полуоктете записывается десятичная цифра адреса, при необходимости поле адреса дополняется нулями до целого числа октетов. Для пакетов установления связи кадры имеют формат, показанный на рис. 4.3.2.4.

Рис. 4.3.2.4. Формат кадра запроса на соединение и соединение установлено

Поле опции содержит целое число октетов, но не более 109, следующее же поле может содержать до 128 байт. Опция типа fast select позволяет поместить до 64 байтов в информационном поле пользователя, во многих случаях этого оказывается достаточно и исключается необходимость переходить в режим пересылки данных.

Если вызываемое DTE не присылает сообщения вызов принят или запрос завершения (установление связи отвергнуто) за отведенное для этого время, процедура завершается и процессу, инициализировавшему запрос, присылается соответствующий код ошибки. При успешной обработке запроса (прислано сообщение соединение установлено) система переходит в режим обмена данными. DTE может в любой момент инициировать процедуру разрыва связи, послав сообщение запрос завершения. DCE сообщает о завершении соединения путем присылки пакета индикация завершения, на который DTE должно прислать отклик подтверждение завершения. Формат пакетов запроса и подтверждения завершения отображен на рис. 4.3.2.4. и 4.3.2.5. Байты 1 и 2 на рисунке 4.3.2.5 не показаны, так как они идентичны тому, что представлено на рис. 4.3.2.4.

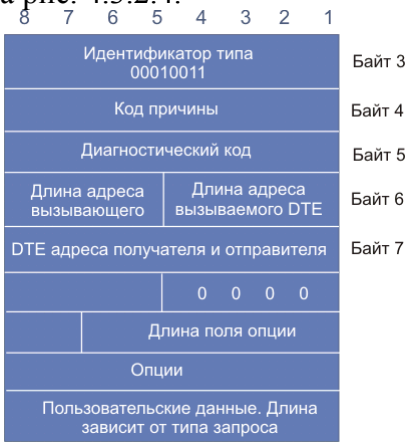


Рис. 4.3.2.5. Формат пакетов запроса завершения

Коды причины завершения связи приведены в таблице 4.3.2.1. Однобайтовое поле диагностический код позволяет уточнить причину. В таблице 4.3.2.4 приведены коды причины повторного пуска. Формат пакетов подтверждения завершения представлен на рис. 4.3.2.6.

Таблица 4.3.2.4. Коды причин повторного пуска

Код причины	**Причина повторного пуска**
0x1	Ошибка локальной процедуры
0x3	Перегрузка сети
0x7	Сеть работоспособна

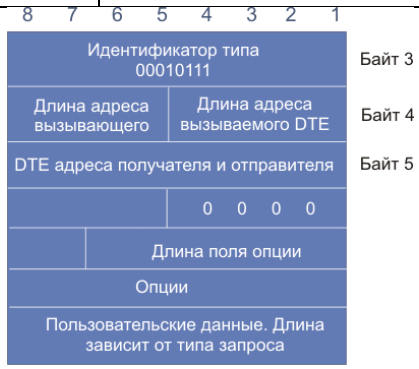


Рис. 4.3.2.6. Формат пакетов подтверждения завершения

Для инициализации обмена информацией (первичного или повторного), а также для прерывания виртуальной связи и возвращения виртуальных каналов в исходное состояние используются запросы повторного пуска (и подтверждение повторного пуска). DTE может выдать запрос повторного пуска (к DCE) в любой момент времени, переводя логический канал в исходное состояние. DCE в ответ должно послать сообщение подтверждение повторного пуска. Инициатором повторного пуска может быть и dce, для этого оно посылает сообщение индикация повторного пуска. DTE в результате устанавливает логический канал в исходное

состояние и посылает dce сообщение подтверждение повторного пуска. Форматы пакетов, несущих эти сообщения показаны на рис. 4.3.2.6 и 4.3.2.7. Эти пакеты не имеют полей группового номера логического канала и LCN (см. рис. 4.3.2.7 и .8). Процедура повторной установки во многом аналогична повторному пуску и используются всякий раз при выявлении сбоя, чтобы вернуть виртуальную связь или постоянный виртуальный канал в исходное состояние.

Рис. 4.3.2.7. Формат пакета запроса повторного пуска (слева) и повторной установки

Таблица 4.3.2.5. Коды причин повторной установки

Причина повторной установки	**Код причины**
Установка по инициативе dte	0x0
Повреждение постоянного виртуального канала	0x1
Ошибка при исполнении удаленной процедуры	0x3
Ошибка при выполнении локальной процедуры	0x5
Перегрузка сети	0x7
Удаленное DTE работоспособно (постоянный виртуальный канал)	0x9
Сеть работоспособна (постоянный виртуальный канал)	0xf
Несовместимость партнеров	0x11

Партнер - получатель этого запроса должен прислать сообщение подтверждение повторной установки (рис. 4.3.2.8). При этом возможны потери информации (также как и в случае повторного пуска), так как некоторые пакеты, находящиеся в сети в момент реализации запроса повторной установки или повторного пуска будут потеряны.

Инициатором посылки запроса повторной установки может быть dte и dce. Коды причин повторной установки представлены в таблице 4.3.2.5.

Рис. 4.3.2.8. Формат пакета подтверждения повторного пуска (слева) и повторной установки (справа)

Пакеты данных передаются по постоянным виртуальным каналам или через виртуальные соединения после их создания. Пакеты данных распознаются по нулевому младшему биту (бит с номером 1) в третьем октете. Остальные биты этого октета используются для управления. Форматы пакетов данных показаны на рис. 4.3.2.9.

Информационное поле начинается с четвертого байта (при расширенной нумерации с пятого) и может иметь длину 16-4096, хотя в рекомендациях стандарта x.25 оговорена величина 128 октетов. Если принимающая сторона не способна принять пакет данной длины, связь должна быть переустановлена, а стороне-инициатору соединения послано сообщение об

ошибке. Каждому пакету данные присваивается порядковый номер N(S), значение которого при установлении соединения равно нулю.

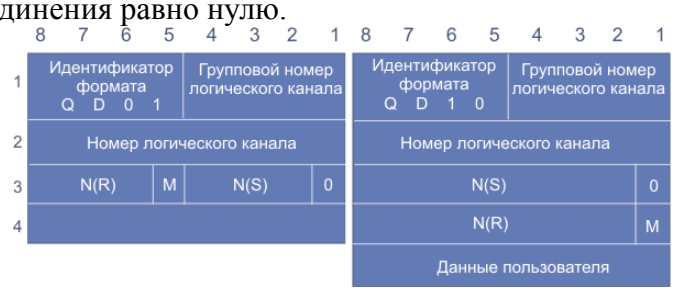


Рис. 4.3.2.9. Форматы пакетов данные. Слева - по модулю 8, справа - по модулю 128

Q -бит определяет тип кадра-пакета, Q=1 - управляющий пакет для PAD, Q=0 - информационный пакет. Бит D используется для запроса специального отклика на пакет со стороны удаленного конца виртуального канала. Бит M указывает на то, что данный пакет является частью более крупного пакета, который должен быть воссоздан позднее.

Индекс S (send) соответствует отправке, а индекс R - приему (receive). Если используется нумерация пакетов по модулю 8, N(S) занимает биты 2-4 включительно, при нумерации по модулю 128 для этого отводятся биты 2-8. Нумерация пакетов позволяет выявить потерю пакетов или изменение порядка их доставки. N(R) является номером пакета с принимающей стороны. Бит подтверждения доставки D (идентификатор формата) служит для указания необходимости сообщения о доставке данных получателем. Если D=1, то DTE обязано подтвердить доставку. Обязательность процедуры подтверждения определяется уже на фазе установления связи (сообщение запрос на установление связи принят). Если какой-либо узел по пути пересылки пакета не поддерживает процедуру подтверждения доставки, он пошлет сообщение запрос завершения (причина - несовместимость у адресата) и связь должна быть сформирована заново с учетом необходимости подтверждения во всех узлах-участниках. Размер поля данные в пакете может быть разным для разных узлов, участвующих в обмене. По этой причине число полученных пакетов может оказаться больше (или меньше) числа посланных. Для таких случаев предусмотрен флаг m (дополнительные данные). Возможность фрагментации и последующей сборки пакетов определяется управляющими битами M и D (см. таблицу 4.3.2.6).

Таблица 4.3.2.6. Управление фрагментацией и сборкой пакетов с помощью битов M и D

Бит m	**Бит d**	**Выполнение объединения с последующим пакетом (реализуется сетью)**
0	0	Нет
0	1	Нет
1	0	Да
1	1	Нет

Таким образом, при фрагментации исходного сообщения все пакеты кроме последнего должны иметь бит m=1. Нумерация пакетов по модулю 8 означает, что им последовательно присваиваются номера 0,1,2,3,4,5,6,7,0,1,2 и т.д. Аналогично при нумерации по модулю 128 - 0,1,2,...127,0,1,2,3 и т.д. Форма нумерации пакетов определяет также размер “окна”, то есть число пакетов, которые могут быть переданы, не дожидаюсь подтверждения получения. По умолчанию размер окна равен 2, другие значения могут быть согласованы на фазе установления соединения. Принцип использования окон при передаче пакетов более подробно описан в разделе “3.6.2 Протокол TCP”.

Для управления процессом передачи данных используются сообщения “готов к приему” и “не готов к приему”. Форматы этих пакетов показаны на рис. 4.3.2.10 и 4.3.2.11.

Рис 4.3.2.10. Формат пакетов готовность к приему и неготовность к приему при нумерации по модулю 8.

Рис 4.3.2.11. Формат пакетов готовность к приему и неготовность к приему при нумерации по модулю 128.

Код N(R) на входе DCE должен лежать в пределах между N(R) последнего принятого пакета и N(S) следующего пакета, который должен быть послан из DCE к DTE. При невыполнении этого условия связь будет переустановлена и передача повторена. Пакеты готовность к приему используются для сообщения о готовности принять пакеты, с номерами, начиная с номера N(R), приведенного в пакете. Пакеты неготовность к приему служат для того, чтобы сообщить о временной неспособности принять данные. При поступлении этого сообщения отправитель должен прервать передачу до получения сообщения готовность к приему. DTE может передавать данные удаленному DTE, не следуя правилам управления потоком данных. Для реализации такой возможности предусмотрена операция прерывания. Эта операция не влияет на передачу данных и управление. Формат пакета прерывание и подтверждение прерывания показан на рис. 4.3.2.12.

Рис. 4.3.2.12. Формат пакетов прерывание и подтверждение прерывания

Идентификатор формата равен 0x1 для нумерации по модулю 8 и 0x2 при нумерации по модулю 128. Передав сообщение прерывание, DTE должно ожидать получение пакета подтверждение прерывания. Максимальный размер поля данные пользователя в пакете прерывание не должен превышать 32 байт.

Иногда в сетях для сообщения об ошибке используется пакет “диагностика”. Этот пакет посылается DCE, адресуется DTE и несет информацию о неустранимых на уровне пакетов ошибках. Пакет диагностика посылается лишь один раз сразу после выявления ошибки. Подтверждения его получения не требуется. Формат пакета показан на рис. 4.3.2.13.

Рис. 4.3.2.13. Формат пакета диагностика

Поле код диагностики несет в себе информацию об ошибке, вызвавшей посылку этого пакета. Если же пакет диагностика передается в качестве отклика на пакет с ошибками от DTE, то поле уточнение диагностики содержит первые три байта пакета DTE.

Современные сети создаются ради доступа к определенным услугам. В протоколе X.25 предусмотрена процедура, которая позволяет получить текущие значения параметров услуг (опций) и модифицировать их. Эта процедура называется регистрацией и не является обязательной. Форматы пакетов запроса регистрации и подтверждения регистрации показаны на рис. 4.3.2.14 и 4.3.2.15. Максимальный размер поля регистрация составляет 109 байт. Инициатором регистрации всегда является dte, которое передает запрос регистрации. В качестве отклика dce посылает пакет подтверждение регистрации, в котором содержатся информация о параметрах доступных услуг. Для выявления доступных услуг может быть послан запрос регистрации, не содержащий списка запрашиваемых услуг.

Рис. 4.3.2.14. Формат пакетов запрос регистрации

Рис. 4.3.2.15. Формат пакетов подтверждение регистрации

Получив список доступных услуг из сообщения подтверждение регистрации, может поменять параметры некоторых из них. Если значение какого-либо параметра услуги (опции) не разрешено, DCE должно сообщить разрешенное значение параметра и максимальное и или минимальное разрешенное значение (в зависимости от того больше или меньше допустимого оказалось значение запрошенного параметра).

Неисправность сети может привести к тому, что та или иная согласованная ранее услуга станет недоступной. В этом случае DCE должно инициировать процедуру повторного пуска, чтобы уведомить DTE о случившихся изменениях.

Кроме процедуры регистрации к необязательным процедурам относятся услуги для замкнутой группы, идентификация пользователей сети, группа поиска, ускоренный обмен, переадресация вызовов, выбор транзитной сети, сообщения о модифицированном адресе, согласование параметров управления потоком и некоторые другие.

Повторная передача пакетов согласуется на определенное время и может использоваться во всех логических каналах DTE-DCE. DTE запрашивает повторную передачу одного или нескольких пакетов данные путем посылки сообщения отказ reject), которое определяет логический канал и порядковый номер пакета N(R). Получив пакет отказ DCE, DTE начинает повторную передачу пакетов. Формат пакетов отказ для случаев нумерации по модулю 8 и 128 показан на рис. 4.3.2.16.

Рис. 4.3.2.16. Форматы пакетов типа отказ для нумерации по модулю 8 (слева) и 128

Программное обеспечение принимающей и передающей сторон должно иметь переменные состояния V(R) и V(S), содержащие, соответственно, номера пакетов, которые предстоит получить и послать (см. описание процедуры HDLC). После посылки очередного пакета с N(S) значение V(S) увеличивается на 1. Принимающая сторона сравнивает V(R) с N(S) полученного пакета, при совпадении укладывает N(S) в поле N(R) пакета-отклика и инкрементирует V(R). Отправитель при получении пакета проверяет равенство переменной V(S) и кода поля N(R) в пакете-отклике. Если при получении пакета выясняется, что V(R) не равно N(S), V(R) не инкрементируется, а принимающая сторона отправляет отклик с N(R)=V(R). Отправитель, получив этот отклик и обнаружив, что V(S) не равно N(R), узнает о происшедшем сбое. Номер логического канала (LCN) служит для того, чтобы определить соответствие межу DTE и местным DCE. LCN вместе с полем группового номера логического канала занимают 12 бит, что позволяет иметь до 4095 логических каналов (LCN=0 зарезервировано для использования DCE).

4 бита первого байта управляющего пакета содержат в себе код типа сообщения (таблица 4.3.2.7):

Таблица 4.3.2.7 Коды типов сообщений

Код типа сообщения	**Команда PAD**	**Отправитель**
0001	Команда разъединения	ЭВМ
0010	Установление параметров	ЭВМ
0011	Индикация разъединения	ЭВМ или PAD
0100	Чтение параметров	ЭВМ
0101	Ошибка	PAD
0110	Установка и чтение параметров	ЭВМ

В поле управляющего сообщения PAD может быть включено любое число параметров, которое допускает максимальный размер пакета. Каждый параметр имеет свой код-номер, за которым в пакете следует значение параметра (таблица 4.3.2.8):

Таблица 4.3.2.8. Коды параметров PAD

Код параметра	**Описание**
1	Обращение к ПАД с использованием управляющего символа
2	Эхо-контроль

3	Выбор сигнала посылки пакета
4	Выбор продолжительности ожидания для таймера
5	Управление вспомогательным устройством
6	Подавление управляющих сигналов ПАД
7	Выбор действий ПАД при получении сигнала разрыва
8	Прерывание вывода
9	Кодовая последовательность после сигнала "возврат каретки"
10	Перенос строки, длина которой ограничена размерами экрана дисплея
11	Скорость работы старт-стопного терминала
12	Управление потоком ПАД
13	Вставка символа "перевод строки" после символа "возврат каретки"
14	Заполнение после сигнала "перевод строки"
15	Редактирование
16	Стирание символа
17	Стирание строки
18	Вывод строки на экран дисплея
19	Редактирование сигналов управления ПАД
20	Маскирование эхо-контроля
21	Обработка символов контроля на четность
22	Ожидание страницы

При работе TCP/IP сети через каналы X.25 и наоборот следует учитывать некоторые отличия кодов предпочтения в полях TOS. Таблица 4.3.2.9 содержит соответствие этих кодов для этих сетей.

Таблица 4.3.2.9. Соответствие кодов TOS для сетей TCP/IP и X.25

IP	**X.25**	**IP**	**X.25**
000	00	001	01
010	10	011 - 111	11

Для реализации работы сетей ISDN по существующим каналам сети X.25 разработан протокол X.31. X.31 организует канал пользователь-маршрутизатор X.25 (через посредство ISDN) и регламентирует работу ISDN с пакетами X.25.

Для решения первой задачи используется сообщение SETUP. Вторая задача решается, когда канал до маршрутизатора сформирован. На этом этапе привлекается набор протоколов X.25, возможно применение протокола X.75 (ISO 8208), который является расширением X.25 для межсетевых связей.

14. Понятие MAC адреса, его структура.

MAC адрес (Media Access Control) - это уникальный шестнадцатиричный серийный номер, назначаемый каждому сетевому устройству Ethernet, для идентификации его в сети. Для сетевых устройств (так же как и для большинства других сетевых типов) этот адрес устанавливается во время изготовления, хотя обычно, он может быть изменен при помощи соответствующей программы.

Каждая сетевая карта имеет уникальный MAC адрес, таким образом она может эксклюзивно забирать пакеты из сетевого провода, предназначенные для нее. Если MAC адрес не является единственным, то не существует способа провести различие между двумя станциями. Устройства в сети просматривают сетевой трафик и ищут свой MAC адрес в каждом пакете, чтобы определить, должны ли они декодировать этот пакет или нет. Существуют специальные способы для широковещательной рассылки сообщений каждому устройству.

MAC адреса имеют длину 6 байт и обычно записываются шестнадцатиричным числом в виде 12:34:56:78:90:AB (двоеточия могут отсутствовать,но их наличие делает адрес более читабельным). Каждый производитель Ethernet устройств использует определенный диапазон MAC адресов, который ему отведен. первые три байта адреса определяют производителя.

Структура 48-битного стандартного адреса

Чтобы распределить возможные диапазоны адресов между многочисленными изготовителями сетевых адаптеров, была предложена следующая структура адреса

- Младшие 24 разряда кода адреса называются OUA (Organizationally Unique Address) - организационно уникальный адрес. Именно их присваивает производитель сетевого адаптера. Всего возможно свыше 16 миллионов комбинаций.
- Следующие 22 разряда кода называются OUI (Organizationally Unique Identifier) - организационно уникальный идентификатор. IEEE присваивает один или несколько ОШ каждому производителю сетевых адаптеров. Это позволяет исключить совпадения адресов адаптеров от разных производителей. Всего возможно свыше 4 миллионов разных OUI. Вместе OUA и OUI называются UAA (Universally Administered Address) - универсально управляемый адрес или IEEE-адрес.
- Два старших разряда адреса являются управляющими и определяют тип адреса, способ интерпретации остальных 46 разрядов. Старший бит I/G (Individual/Group) определяет, индивидуальный это адрес или групповой. Если он установлен в 0, то мы имеем дело с индивидуальным адресом, если установлен в 1, то с групповым (многопунктовым или функциональным) адресом. Пакеты с групповым адресом получают все имеющие его сетевые адаптеры, причем групповой адрес определяется всеми 46 младшими разрядами. Второй управляющий бит U/L (Universal/Local) называется флажком универсального/местного управления и определяет, как был присвоен адрес данному сетевому адаптеру. Обычно он установлен в 0. Установка бита U/L в 1 означает, что адрес задан не производителем сетевого адаптера, а организацией, использующей данную сеть. Это довольно редкая ситуация.

Для широковещательной передачи используется специально выделенный сетевой адрес, все 48 битов которого установлены в единицу. Его принимают все абоненты сети независимо от их индивидуальных и групповых адресов.

Данной системы адресов придерживаются, например, такие популярные сети, как Ethernet, Fast Ethernet, Token-Ring, FDDI, lOOVG-AnyLAN. Ее недостатки - высокая сложность аппаратуры сетевых адаптеров, а также большая доля служебной информации в передаваемом пакете (адрес источника и адрес приемника требуют уже 96 битов пакета, или 12 байт).

15. Протокол маршрутизации BGP

Протокол BGP (RFC-1267, BGP-3; RFC-1268; RFC-1467, BGP-4; -1265-66, 1655) разработан компаниями IBM и CISCO. Главная цель BGP - сократить транзитный трафик. Местный трафик либо начинается, либо завершается в автономной системе (AS); в противном случае - это транзитный трафик. Системы без транзитного трафика не нуждаются в BGP (им достаточно EGP для общения с транзитными узлами). Но не всякая ЭВМ, использующая протокол BGP, является маршрутизатором, даже если она обменивается маршрутной информацией с пограничным маршрутизатором соседней автономной системы. AS передает информацию только о маршрутах, которыми она сама пользуется. BGP-маршрутизаторы обмениваются сообщениями об изменении маршрутов (UPDATE-сообщения, рис. 4.4.11.4.1). Максимальная длина таких сообщений составляет 4096 октетов, а минимальная 19 октетов. Каждое сообщение имеет заголовок фиксированного размера. Объем информационных полей зависит от типа сообщения.

Рис. 4.4.11.4.1. Формат BGP-сообщений об изменениях маршрутов

Поле маркер содержит 16 октетов и его содержимое может легко интерпретироваться получателем. Если тип сообщения "OPEN", или если код идентификации в сообщении open равен нулю, то поле маркер должно быть заполнено единицами. Маркер может использоваться для обнаружения потери синхронизации в работе BGP-партнеров. Поле длина имеет два октета и определяет общую длину сообщения в октетах, включая заголовок. Значение этого поля должно лежать в пределах 19-4096. Поле тип представляет собой код разновидности сообщения и может принимать следующие значения:

1	OPEN	(открыть)
2	UPDATE	(изменить)
3	NOTIFICATION	(внимание)
4	KEEPALIVE	(еще жив)

После того как связь на транспортном протокольном уровне установлена, первое сообщение, которое должно быть послано - это OPEN. При успешном прохождении этого сообщения партнер должен откликнуться сообщением KEEPALIVE ("Еще жив"). После этого возможны любые сообщения. Кроме заголовка сообщение open содержит следующие поля (рис. 4.4.11.4.2):

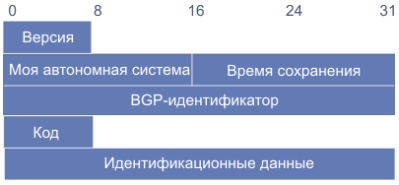


Рис. 4.4.11.4.2 Формат сообщения open

Поле версия описывает код версии используемого протокола, на сегодня для BGP он равен 4. Двух-октетное поле моя автономная система определяет код AS отправителя. Поле время сохранения характеризует время в секундах, которое отправитель предлагает занести в таймер сохранения. После получения сообщения OPEN BGP-маршрутизатор должен выбрать значение времени сохранения. Обычно выбирается меньшее из полученного в сообщении open и значения, определенного при конфигурации системы (0-3сек). Время сохранения определяет максимальное время в секундах между сообщениями KEEPALIVE и UPDATE или между двумя UPDATE-сообщениями. Каждому узлу в рамках BGP приписывается 4-октетный идентификатор (BGP-identifier, задается при инсталляции и идентичен для всех интерфейсов локальной сети). Если два узла установили два канала связи друг с другом, то

согласно правилам должен будет сохранен канал, начинающийся в узле, BGP-идентификатор которого больше. Предусмотрен механизм разрешения проблемы при равных идентификаторах.
Одно-октетный код идентификации позволяет организовать систему доступа, если он равен нулю, маркер всех сообщений заполняется единицами, а поле идентификационных данных должно иметь нулевую длину. При неравном нулю коде идентификации должна быть определена процедура доступа и алгоритм вычисления кодов поля маркера. Длина поля идентификационных данных определяется по формуле:
Длина сообщения = 29 + длина поля идентификационных данных.
Минимальная длина сообщения open составляет 29 октетов, включая заголовок.
Сообщения типа UPDATE (изменения) используются для передачи маршрутной информации между BGP-партнерами. Этот тип сообщения позволяет сообщить об одном новом маршруте или объявить о закрытии группы маршрутов, причем объявление об открытии нового и закрытии старых маршрутов возможно в пределах одного сообщения. Сообщение UPDATE всегда содержит стандартный заголовок и может содержать другие поля в соответствии со схемой:

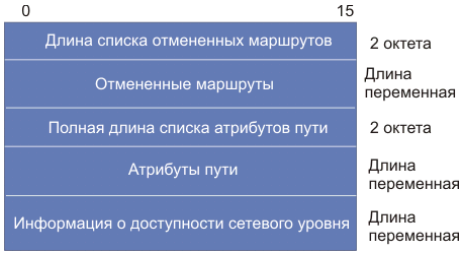
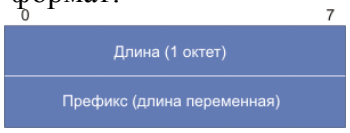


Рис. 4.4.11.4.3 Формат update-сообщения

Если длина списка отмененных маршрутов равна нулю, ни один маршрут не отменен, а поле отмененные маршруты в сообщении отсутствует. Поле отмененные маршруты имеет переменную длину и содержит список IP-адресных префиксов маршрутов, которые стали недоступны. Каждая такая запись имеет формат:



Длина префикса (в битах), равная нулю означает, что префикс соответствует всем IP-адресам, а сам имеет нулевой размер. Поле префикс содержит IP-адресные префиксы, за которыми следуют разряды, дополняющие их до полного числа октетов. Значения этих двоичных разрядов смысла не имеют.
Нулевое значение полной длины списка атрибутов пути говорит о том, что информация о доступности сетевого уровня в UPDATE-сообщении отсутствует. Список атрибутов пути присутствует в любом UPDATE-сообщении. Этот список имеет переменную длину, а каждый атрибут содержит три составные части: тип атрибута, длину атрибута и значение атрибута. Тип атрибута представляет собой двух-октетное поле со структурой:

Старший бит (бит0) поля флаги атрибута определяет, является ли атрибут опционным (бит0=1) или стандартным (well-known, бит0=0). Бит 1 этого поля определяет, является ли атрибут переходным (бит1=1) или непереходным (бит1=0). Для обычных атрибутов этот бит должен быть равен 1. Третий бит (бит 2) поля Флагов атрибута определяет, является ли информация в опционном переходном атрибуте полной (бит2=0) или частичной (бит2=1). Для обычных и для опционных непереходных атрибутов этот бит должен быть равен нулю. Бит 3 поля флагов атрибута информирует о том, имеет ли длина атрибута один (бит3=0) октет или два октета (бит3=1). Бит3 может быть равен 1 только в случае, когда длина атрибута более 255 октетов. Младшие 4 бита октета флагов атрибута не используются (и должны обнулять-

ся). Если бит3=0, то третий октет атрибута пути содержит длину поля данных атрибута в октетах. Если же бит3=1, то третий и четвертый октеты атрибута пути хранят длину поля данных атрибута. Остальные октеты поля атрибут пути характеризуют значение атрибута и интерпретируются согласно флагам атрибута.

Атрибуты пути бывают "стандартные обязательные" (well-known mandatory), "стандартные на усмотрение оператора", "опционные переходные" и "опционные непереходные". Стандартные атрибуты должны распознаваться любыми BGP-приложениями. Опционные атрибуты могут не распознаваться некоторыми приложениями. Обработка нераспознанных атрибутов задается битом 1 поля флагов. Пути с нераспознанными переходными опционными атрибутами должны восприниматься, как рабочие. Один и тот же атрибут может появляться в списке атрибутов пути только один раз.

Предусмотрены следующие разновидности кодов типа атрибута:

ORIGIN (код типа = 1) - стандартный обязательный атрибут, который определяет происхождение путевой информации. Генерируется автономной системой, которая является источником маршрутной информации. Значение атрибута в этом случае может принимать следующие значения:

Код атрибута	**Описание**
0	IGP - информация достижимости сетевого уровня является внутренней по отношению к исходной автономной системе;
1	EGP - информация достижимости сетевого уровня получена с помощью внешнего протокола маршрутизации;
2	Incomplete - информация достижимости сетевого уровня получена каким-то иным способом.

AS_PATH (код типа = 2) также является стандартным обязательным атрибутом, который составлен из совокупности сегментов пути. Атрибут определяет автономные системы, через которые доставлена маршрутная информация. Когда BGP-маршрутизатор передает описание маршрута, которое он получил от своего BGP-партнера, он модифицирует AS_PATH-атрибут, соответствующий этому маршруту, если информация передается за пределы автономной системы. Каждый сегмент AS_PATH состоит из трех частей <тип сегмента пути, длина сегмента пути и оценка сегмента пути>. Тип сегмента пути представляет в свою очередь однооктетное поле, которое может принимать следующие значения:

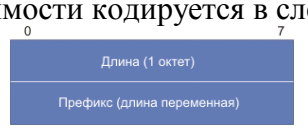
Код типа сегмента	**Описание**
1	AS_set: неупорядоченный набор маршрутов в update сообщении;
2	AS_sequence: упорядоченный набор маршрутов автономной системы в UPDATE-сообщении.

Длина сегмента пути представляет собой одно-октетное поле, содержащее число as, записанных в поле оценка сегмента пути. Последнее поле хранит один или более кодов автономной системы, по два октета каждый.

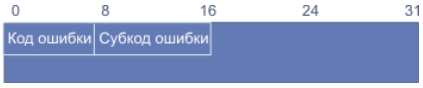
NEXT_HOP (код типа = 3) - стандартный обязательный атрибут, определяющий IP-адрес пограничного маршрутизатора, который должен рассматриваться как цель следующего шага на пути к точке назначения.

MULTI_EXIT_DISC (код типа = 4) представляет собой опционный непереходной атрибут, который занимает 4 октета и является положительным целым числом. Величина этого атрибута может использоваться при выборе одного из нескольких путей к соседней автономной системе.

LOCAL_PREF (код типа = 5) является опционным атрибутом, занимающим 4 октета. Он используется BGP-маршрутизатором, чтобы сообщить своим BGP-партнерам в своей собственной автономной системе степень предпочтения объявленного маршрута.
ATOMIC_AGGREGATE (код типа = 6) представляет собой стандартный атрибут, который используется для информирования партнеров о выборе маршрута, обеспечивающего доступ к более широкому списку адресов.
aggregator (код типа = 7) - опционный переходной атрибут с длиной в 6 октетов. Атрибут содержит последний код автономной системы, который определяет агрегатный маршрут (занимает два октета), и IP-адрес BGP-маршрутизатора, который сформировал этот маршрут (4 октета). Объем информации о достижимости сетевого уровня равен (в октетах):
Длина сообщения UPDATE - 23 - полная длина атрибутов пути - длина списка отмененных маршрутов. Информация о достижимости кодируется в следующей форме:



Поле длина определяет длину IP-адресного префикса в битах. Если длина равна нулю, префикс соответствует всем IP-адресам. Префикс содержит IP-адресные префиксы и двоичные разряды, дополняющие код до целого числа октетов.
Информация о работоспособности соседних маршрутизаторов получается из KEEPALIVE-сообщений, которые должны посылаться настолько часто, чтобы уложиться во время, отведенное таймером сохранения (hold). Обычно это время не должно превышать одной трети от времени сохранения, но не должно быть и меньше 1 секунды. Если выбранное значение времени сохранения равно нулю, периодическая посылка KEEPALIVE-сообщений не обязательна.
NOTIFICATION-сообщения посылаются, когда обнаружена ошибка. BGP-связь при этом немедленно прерывается. Помимо заголовка NOTIFICATION-сообщение имеет следующие поля:



Код ошибки представляет собой одно-октетное поле и указывает на тип данного сообщения. Возможны следующие коды ошибки:

Таблица 4.4.11.4.1. Коды ошибок

Код ошибки	**Описание**
1	Ошибка в заголовке сообщения.
2	Ошибка в сообщении open
3	Ошибка в сообщении update
4	Истекло время сохранения
5	Ошибка машины конечных состояний
6	Прерывание

При отсутствии фатальной ошибки BGP-партнер может в любой момент прервать связь, послав NOTIFICATION-сообщение с кодом ошибки прерывание.
Одно-октетное поле субкод ошибки предоставляет дополнительную информацию об ошибке. Каждый код ошибки может иметь один или более субкодов. Если поле содержит нуль, это означает, что никаких субкодов не определено.

Таблица 4.4.11.4.2 Субкоды ошибок

Ошибка	Субкод	Описание
Заголовок	1 2 3	Соединение не синхронизовано Неверная длина сообщения Неверный тип сообщения
Сообщения OPEN	1 2 3 4 5 6	Неверный код версии Ошибочный код as-партнера Ошибочный идентификатор BGP Ошибка в коде идентификации Ошибка при идентификации Неприемлемое время сохранения
Сообщения UPDATE	1 2 3 4 5 6 7 8 9 10 11	Ошибка в списке атрибутов Не узнан стандартный атрибут Отсутствует стандартный атрибут Ошибка в флагах атрибута Ошибка в длине атрибута Неправильный атрибут origin Циклический маршрут Ошибка в атрибуте next_hop Ошибка в опционном атрибуте Ошибка в сетевом поле Ошибка в as_path

Вся маршрутная информация хранится в специальной базе данных RIB (routing information base). Маршрутная база данных BGP состоит из трех частей:
1.ADJ-RIBS-IN: Запоминает маршрутную информацию, которая получена из update-сообщений. Это список маршрутов, из которого можно выбирать. (policy information base - PIB).
2. LOC-RIB: Содержит локальную маршрутную информацию, которую BGP-маршрутизатор отобрал, руководствуясь маршрутной политикой, из ADJ-RIBS-IN.
3. ADJ-RIBS-OUT: Содержит информацию, которую локальный BGP-маршрутизатор отобрал для рассылки соседям с помощью UPDATE-сообщений.

Так как разные BGP-партнеры могут иметь разную политику маршрутизации, возможны осцилляции маршрутов. Для исключения этого необходимо выполнять следующее правило: если используемый маршрут объявлен не рабочим (в процессе корректировки получено сообщение с соответствующим атрибутом), до переключения на новый маршрут необходимо ретранслировать сообщение о недоступности старого всем соседним узлам.
Протокол BGP позволяет реализовать маршрутную политику, определяемую администратором AS (см. раздел "Автономные системы и маршрутная политика"). Политика отражается в конфигурационных файлах BGP. Маршрутная политика это не часть протокола, она определяет решения, когда место назначения достижимо несколькими путями, политика отражает соображения безопасности, экономические интересы и пр. Количество сетей в пределах одной AS не лимитировано. Один маршрутизатор на много сетей позволяет минимизировать таблицу маршрутов.
BGP использует три таймера:
Connectretry (сбрасывается при инициализации и коррекции; 120 сек), **Holdtime** (запускается при получении команд Update или Keepalive; 90сек) и **keepalive** (запускается при посылке сообщения Keepalive; 30сек).

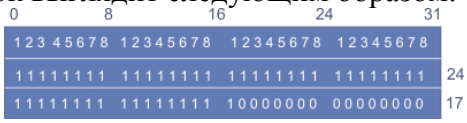
BGP отличается от RIP и OSPF тем, что использует TCP в качестве транспортного протокола. Две системы, использующие BGP, связываются друг с другом и пересылают посредством TCP полные таблицы маршрутизации. В дальнейшем обмен идет только в случае каких-то изменений. ЭВМ, использующая BGP, не обязательно является маршрутизатором. Сообщения обрабатываются только после того, как они полностью получены.
BGP является протоколом, ориентирующимся на вектор расстояния. Вектор описывается списком AS по 16 бит на AS. BGP регулярно (каждые 30сек) посылает соседям TCP-сообщения, подтверждающие, что узел жив (это не тоже самое что "Keepalive" функция в TCP). Если два BGP-маршрутизатора попытаются установить связь друг с другом одновременно, такие две связи могут быть установлены. Такая ситуация называется столкновением, одна из связей должна быть ликвидирована. При установлении связи маршрутизаторов сначала делается попытка реализовать высший из протоколов (например, BGP-4), если один из них не поддерживает эту версию, номер версии понижается.
Протокол BGP-4 является усовершенствованной версией (по сравнению с BGP-3). Эта версия позволяет пересылать информацию о маршруте в рамках одного IP-пакета. Концепция классов сетей и субсети находятся вне рамок этой версии. Для того чтобы приспособиться к этому, изменена семантика и кодирование атрибута AS_PASS. Введен новый атрибут LOCAL_PREF (степень предпочтительности маршрута для собственной AS), который упрощает процедуру выбора маршрута. Атрибут INTER_AS_METRICSпереименован в MULTI_EXIT_DISC (4 октета; служит для выбора пути к одному из соседей). Введены новые атрибуты ATOMIC_AGGREGATE и AGGREGATOR, которые позволяют группировать маршруты. Структура данных отражается и на схеме принятия решения, которая имеет три фазы:

1. Вычисление степени предпочтения для каждого маршрута, полученного от соседней AS, и передача информации другим узлам местной AS.

2. Выбор лучшего маршрута из наличного числа для каждой точки назначения и укладка результата в LOC-RIB.

3. Рассылка информации из loc_rib всем соседним AS согласно политике, заложенной в RIB. Группировка маршрутов и редактирование маршрутной информации.

Бесклассовая интердоменная маршрутизация (CIDR- classless interdomain routing, RFC-1520, -1519) - способ избежать того, чтобы каждая C-сеть требовала свою таблицу маршрутизации. Основополагающий принцип CIDR заключается в группировке (агрегатировании) IP-адресов таким образом, чтобы сократить число входов в таблицах маршрутизации (RFC-1519, RFC-1518, RFC-1467, RFC-1466). Протокол совместим с RIP-2, OSPF и BGP-4. Основу протокола составляет идея бесклассовых адресов, где нет деления между полем сети и полем ЭВМ. Дополнительная информация, например 32-разрядная маска, выделяющая поле адреса сети, передается в рамках протокола маршрутизации. При этом выдерживается строгая иерархия адресов: провайдер > предприятие > отдел/здание > сегмент локальной сети. Групповой (агрегатный) адрес воспринимается маршрутизатором как один адрес. Группу может образовывать только непрерывная последовательность IP-адресов. Такой бесклассовый интернетовский адрес часто называется IP-префиксом. Так адрес 192.1.1.0/24 означает диапазон адресов 192.1.1.0 - 192.1.1.255, а адрес 192.1.128.0/17 описывает диапазон 192.1.128.0 - 192.1.255.255, таким образом, число, следующее после косой черты, задает количество двоичных разрядов префикса. Это представление используется при описании политики маршрутизации и самих маршрутов (см. разд.4.4.11.4 - "Маршрутная политика"). Для приведенных примеров это в терминах масок выглядит следующим образом:



24 и 17 длины префикса сети.

Следует помнить, что маски с разрывами здесь недопустимы. Ниже приведена таблица метрик маршрутизации для различных протоколов.

Протокол	**Метрика**	**Диапазон**	**Код "маршрут недостижим"**
RIP	Число скачков	0-15	16
hello	Задержка в ms	0-29999	30000
BGP	Не определена	0-65534	65535

Колонка "маршрут недостижим" содержит коды метрики, которые говорят о недоступности маршрута. Обычно предполагается, что если послан пакет из точки <A> в точку <B>, то маршруты их в одном и другом направлении совпадают. Но это не всегда так. Пример, когда маршруты пакетов "туда" и "обратно" не совпадают, представлен на рис. 4.4.11.4.4. В предложенной схеме имеется две ЭВМ "Место назначения" и "ЭВМ-отправитель", а также два маршрутизатора "GW-2" и "GW-1".

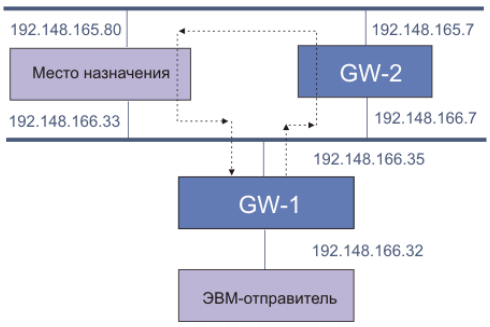


Рис. 4.4.11.4.4. Пример разных маршрутов для пути "туда" и "обратно".

Предполагается, что оператор находится в ЭВМ-отправителе. Команда traceroute 192.148.166.33 в этом случае выдаст:

1 GW-1 (192.148.166.35)

2 Место назначения (192.148.166.33)

Команда же traceroute 192.148.165.80 распечатает:

1 GW-1 (192.148.166.35)

2 GW-2 (192.148.166.7)

3 Место назначения (192.148.165.80)

Команда traceroute -g 192.148.165.80 сообщит вам:

1 GW-1 (192.148.166.35)

2 ***** ; В этом режиме маршрутизатор не откликается

3 Место назначения (192.148.165.80)

4 GW-1 (192.148.166.35)

5 ЭВМ-отправитель (192.148.166.32)

Из приведенных примеров видна также полезность команды traceroute для понимания того, как движутся пакеты в сети. В некоторых случаях это может помочь оптимизировать маршрутизацию и улучшить пропускную способность сети.

Другой полезной командой является Netstat, которая позволяет получить разнообразную информацию о состоянии сети. Существует четыре модификации этой команды:

-a отображает состояния всех соединений;

-i отображает значения конфигурационных параметров;

-r отображает таблицу маршрутов;

-v отображает статистику обмена локального Ethernet-интерфейса.

Например, команда netstat -r может выдать:

Routing tables (таблицы маршрутизации)

Destination	**Gateway**	**Flags**	**Refcnt**	**Use**	**Interface**
Stavropol-GW.ITEP.RU	nb	UGHD	**0**	109	le0
ihep.su	itepgw	UGHD	0	103	le0
m10.ihep.su	itepgw	UGHD	0	16	le0
194.85.66.50	itepgw	UGHD	0	455	le0
Kharkov.ITEP-Kharkov	nb	UGHD	0	105	le0
Bryansk-GW.ITEP.Ru	nb	UGHD	1	8113	le0
193.124.225.67	nb	UGHD	0	0	le0
ixwin.ihep.su	itepgw	UGHD	1	6450	le0
ihep.su	itepgw	UGHD	0	14	le0
192.148.166.21	nb	UGHD	0	109	le0
ihep.su	itepgw	UGHD	0	224	le0
193.124.225.71	nb	UGHD	0	10	le0
194.85.112.0	ITEP-FDDI-BBone.ITEP	UGD	0	253	le0
default	itepgw	UG	10	102497	le0

Здесь приведен только фрагмент маршрутной таблицы. Колонка destination указывает на конечную точку маршрута (default - маршрут по умолчанию), колонка gateway - имена маршрутизаторов, через которые достижим адресат. Флаг "U" (Up) свидетельствует о том, что канал в рабочем состоянии. Флаг "G" указывает на то, что маршрут проходит через маршрутизатор (gateway). При этом вторая колонка таблицы содержит имя этого маршрутизатора. Если флаг "G" отсутствует, ЭВМ непосредственно связана с указанной сетью. Флаг "D" указывает на то, что маршрут был добавлен динамически. Если маршрут связан только с конкретной ЭВМ, а не с сетью, он помечается флагом "H" (host), при этом первая колонка таблицы содержит его IP-адрес. Базовая команда netstat может обеспечить следующую информацию:

Active Internet connections (активные Интернет связи)

Proto	**Recv-Q**	**Send-Q**	**Local Address**	**Foreign Address**	**(state)**
tcp	0	0	127.0.0.1.1313	127.0.0.1.sunrpc	TIME_WAIT
tcp	0	0	ns.1312	193.124.18.65.smtp	SYN_SENT
tcp	0	0	127.0.0.1.1311	127.0.0.1.sunrpc	TIME_WAIT
tcp	0	0	ns.1310	ns.domain	TIME_WAIT
tcp	0	0	127.0.0.1.1309	127.0.0.1.sunrpc	TIME_WAIT
tcp	39	24576	ns.nntp	Bryansk-GW.ITEP.1697	ESTABLISHED
tcp	0	0	ns.telnet	semenov.1802	ESTABLISHED
tcp	0	188	ns.1033	xmart.desy.de.6000	ESTABLISHED
udp	0	0	127.0.0.1.domain	*.*	
udp	0	0	ns.domain	*.*	

16. IP адресация: IPv4, IPv6. Варианты назначения IP адресов.

Один из основных типов пакетов в интернете - IP-пакет (RFC-791), именно он вкладывается в кадр Ethernet и именно в него вкладываются пакеты UDP, TCP. IP-протокол предлагает ненадежную транспортную среду. Ненадежную в том смысле, что не существует гарантии благополучной доставки IP-дейтаграммы. Алгоритм доставки в рамках данного протокола предельно прост: при ошибке дейтаграмма выбрасывается, а отправителю посылается соответствующее ICMP-сообщение (или не посылается ничего). Обеспечение же надежности возлагается на более высокий уровень (UDP или TCP). **Формат IP-пакетов показан на рисунке**

Поле *версия* характеризует версию IP-протокола (например, 4 или 6). Формат пакета определяется программой и, вообще говоря, может быть разным для разных значений поля версия. Только размер и положение этого поля незыблемы. Поэтому в случае изменений длины IP-адреса слишком тяжелых последствий это не вызовет. Понятно также, что значение поля версия во избежании непредсказуемых последствий должно контролироваться программой.

Поле *HLEN* - длина заголовка, измеряемая в 32-разрядных словах, обычно заголовок содержит 20 октетов (HLEN=5, без опций и заполнителя).

Поле *полная длина* определяет полную длину IP-дейтаграммы (до 65535 октетов), включая заголовок и данные.

Поле *тип сервиса* (одно-октетное)(**TOS** - type of service) характеризует то, как должна обрабатываться дейтаграмма. Это поле делится на 6 субполей:

Субполе *Приоритет* предоставляет возможность присвоить код приоритета каждой дейтаграмме (в настоящее время это поле не используется).

Формат поля TOS определен в документе RFC-1349. Биты C, D, T и R характеризуют пожелание относительно способа доставки дейтаграммы. Так D=1 требует минимальной задержки, T=1 - высокую пропускную способность, R=1 - высокую надежность, а C=1 – низкую стоимость. **TOS** играет важную роль в маршрутизации пакетов. Интернет не гарантирует запрашиваемый TOS, но многие маршрутизаторы учитывают эти запросы при выборе маршрута (протоколы OSPF и IGRP). Рекомендуемые значения TOS – открытая информация. Только один бит из четырех в TOS может принимать значение 1.

Поля *идентификатор*, *флаги* (3 бита) и *указатель фрагмента* (fragment offset) управляют процессом фрагментации и последующей "сборки" дейтаграммы. ***Идентификатор*** представляет собой уникальный код дейтаграммы, позволяющий идентифицировать принадлежность фрагментов и исключить ошибки при "сборке" дейтаграмм. Бит 0 поля ***флаги*** является резервным, бит 1 служит для управления фрагментацией пакетов (0 - фрагментация разрешена; 1 - запрещена), бит 2 определяет, является ли данный фрагмент последним (0 – последний фрагмент; 1 - следует ожидать продолжения). Поле *время жизни* (TTL - time to live) задает время жизни дейтаграммы в секундах, т.е. предельно допустимое время пребывания дей-

таграммы в системе. При каждой обработке дейтаграммы, например в маршрутизаторе, это время уменьшается в соответствии со временем пребывания в данном устройстве или согласно протоколу обработки. Если TTL=0, дейтаграмма из системы удаляется. Во многих реализациях TTL измеряется в числе шагов, в этом случае каждый маршрутизатор выполняет операцию TTL=TTL-1. TTL помогает предотвратить зацикливание пакетов. Поле *протокол* аналогично полю *тип* в Ethernet-кадре и определяет структуру поля *данные* (см. табл. 4.4.1.2).

Поле *контрольная сумма заголовка* вычисляется с использованием операций сложения 16-разрядных слов заголовка по модулю 1. Сама контрольная сумма является дополнением по модулю один полученного результата сложения. Обратите внимание, здесь осуществляется контрольное суммирование заголовка, а не всей дейтаграммы. Поле *опции* не обязательно присутствует в каждой дейтаграмме. Размер поля *опции* зависит от того, какие опции применены. Если используется несколько опций, они записываются подряд без каких-либо разделителей. Каждая опция содержит один октет кода опции, за которым может следовать октет длины и серия октетов данных. Если место, занятое опциями, не кратно 4 октетам, используется заполнитель. Структура октета кода опции отражена на рис. 4.4.1.2.

IPv6 представляет собой новую версию протокола Интернет (RFC-1883), являющуюся преемницей версии 4 (IPv4; RFC-791). Изменения IPv6 по отношению к IPv4 можно поделить на следующие группы:

Расширение адресации В IPv6 длина адреса расширена до 128 бит (против 32 в IPv4), что позволяет обеспечить больше уровней иерархии адресации, увеличить число адресуемых узлов, упростить авто-конфигурацию. Для расширения возможности мультикастинг-маршрутизации в адресное поле введено субполе "scope" (группа адресов). Определен новый тип адреса "anycast address" (эникастный), который используется для посылки запросов клиента любой группе серверов. Эникаст адресация предназначена для использования с набором взаимодействующих серверов, чьи адреса не известны клиенту заранее.

Спецификация *формата заголовков* Некоторые поля заголовка IPv4 отбрасываются или делаются опционными, уменьшая издержки, связанные с обработкой заголовков пакетов с тем, чтобы уменьшить влияние расширения длины адресов в IPv6.

Улучшенная поддержка расширений и опций Изменение кодирования опций IP-заголовков позволяет облегчить переадресацию пакетов, ослабляет ограничения на длину опций, и делает более доступным введение дополнительных опций в будущем.

Возможность пометки потоков данных Введена возможность помечать пакеты, принадлежащие определенным транспортным потокам, для которых отправитель запросил определенную процедуру обработки, например, нестандартный тип TOS (вид услуг) или обработка данных в реальном масштабе времени.

Идентификация и защита частных обменов В IPv6 введена спецификация идентификации сетевых объектов или субъектов, для обеспечения целостности данных и при желании защиты частной информации.

Формат и семантика адресов IPv6 описаны в документе RFC-1884. Версия ICMP IPv6 рассмотрена в RFC-1885.

17. Протокол маршрутизации OSPF

Протокол OSPF (Open Shortest Pass First, RFC-1245-48, RFC-1583-1587, алгоритмы предложены Дикстрой) является альтернативой RIP в качестве внутреннего протокола маршрутизации. OSPF представляет собой протокол состояния маршрута (в качестве метрики используется - коэффициент качества обслуживания). Каждый маршрутизатор обладает полной информацией о состоянии всех интерфейсов всех маршрутизаторов (переключателей) автономной системы. Протокол OSPF реализован в демоне маршрутизации gated, который поддерживает также RIP и внешний протокол маршрутизации BGP.

Автономная система может быть разделена на несколько областей, куда могут входить как отдельные ЭВМ, так и целые сети. В этом случае внутренние маршрутизаторы области могут и не иметь информации о топологии остальной части AS. Сеть обычно имеет выделенный (designated) маршрутизатор, который является источником маршрутной информации для остальных маршрутизаторов AS. Каждый маршрутизатор самостоятельно решает задачу оптимизации маршрутов. Если к месту назначения ведут два или более эквивалентных маршрута, информационный поток будет поделен между ними поровну. Переходные процессы в OSPF завершаются быстрее, чем в RIP. В процессе выбора оптимального маршрута анализируется ориентированный граф сети. Ниже описан алгоритм Дикстры по выбору оптимального пути. На иллюстративном рис. 4.2.11.2.1 приведена схема узлов (A-J) со значениями метрики для каждого из отрезков пути. Анализ графа начинается с узла A (Старт). Пути с наименьшим суммарным значением метрики считаются наилучшими. Именно они оказываются выбранными в результате рассмотрения графа (“кратчайшие пути“).

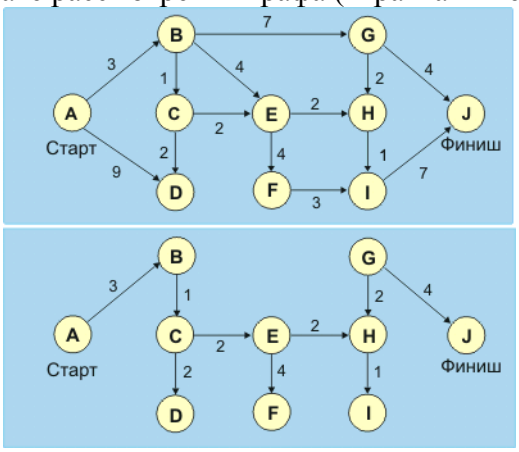


Рис. 4.2.11.2.1 Иллюстрация работы алгоритма Дикстры

Ниже дается формальное описание алгоритма. Сначала вводим некоторые определения.

Пусть **D(v)** равно сумме весов связей для данного пути.

Пусть **c(i,j)** равно весу связи между узлами с номерами **i** и **j.**

Далее следует последовательность шагов, реализующих алгоритм.

1. Устанавливаем множество узлов **N = {1}.**
2. Для каждого узла **v** не из множества **n** устанавливаем **D(v)= c(1,v).**
3. Для каждого шага находим узел **w** не из множества **N**, для которого **D(w)** минимально, и добавляем узел **w** в множество **N.**
4. Актуализируем **D(v)** для всех узлов не из множества **N**
   D(v)=min{D(v), D(v)+c(w,v)}.
5. Повторяем шаги 2-4, пока все узлы не окажутся в множестве **N.**

Топология маршрутов для узла a приведена на нижней части рис. 4.2.11.2.1. В скобках записаны числа, характеризующие метрику отобранного маршрута согласно критерию пункта 3.

Качество сервиса (QoS) может характеризоваться следующими параметрами:

- пропускной способностью канала;
- задержкой (время распространения пакета);
- числом дейтограмм, стоящих в очереди для передачи;

- загрузкой канала;
- требованиями безопасности;
- типом трафика;
- числом шагов до цели;
- возможностями промежуточных связей (например, многовариантность достижения адресата).

Определяющими являются три характеристики: задержка, пропускная способность и надежность. Для транспортных целей OSPF использует IP непосредственно, т.е. не привлекает протоколы UDP или TCP. OSPF имеет свой код (89) в протокольном поле IP-заголовка. Код TOS (type of service) в IP-пакетах, содержащих OSPF-сообщения, равен нулю, значение TOS здесь задается в самих пакетах OSPF. Маршрутизация в этом протоколе определяется IP-адресом и типом сервиса. Так как протокол не требует инкапсуляции пакетов, сильно облегчается управление сетями с большим количеством бриджей и сложной топологией (исключается циркуляция пакетов, сокращается транзитный трафик). Автономная система может быть поделена на отдельные области, каждая из которых становится объектом маршрутизации, а внутренняя структура снаружи не видна (узлы на рис. 4.2.11.2.1 могут представлять собой как отдельные ЭВМ или маршрутизаторы, так и целые сети). Этот прием позволяет значительно сократить необходимый объем маршрутной базы данных. В OSPF используется термин опорной сети (backbone) для коммуникаций между выделенными областями. Протокол работает лишь в пределах автономной системы. В пределах выделенной области может работать свой протокол маршрутизации.

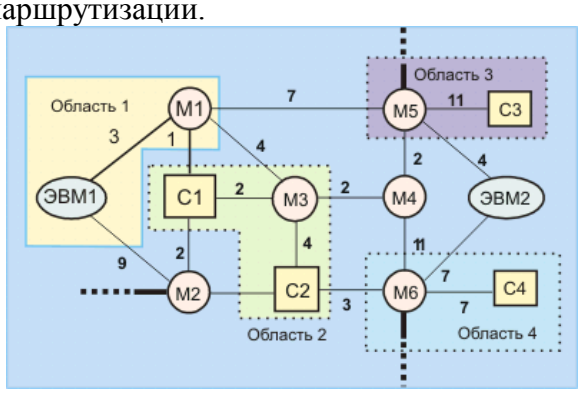


Рис. 4.2.11.2.2 Пример выделения областей при OSPF маршрутизации в автономной системе (M - маршрутизаторы; c - сети).

На рис 4.2.11.2.2 (см. также рис. 4.2.11.2.1) приведен пример выделения областей маршрутизации при OSPF-маршрутизации в пределах автономной системы. На рис. 4.2.11.2.2 маршрутизаторы M4 и M2 выполняют функция опорной сети для других областей. В выделенных областях может быть любое число маршрутизаторов. Более толстыми линиями выделены связи с другими автономными системами.

При передаче OSPF-пакетов фрагментация не желательна, но не запрещается. Для передачи статусной информации OSPF использует широковещательные сообщения Hello. Для повышения безопасности предусмотрена авторизация процедур. OSPF-протокол требует резервирования двух мультикастинг-адресов:

224.0.0.5 предназначен для обращения ко всем маршрутизаторам, поддерживающим этот протокол.

224.0.0.6 служит для обращения к специально выделенному маршрутизатору.

Любое сообщение OSPF начинается с 24-октетного заголовка:

Рис. 4.2.11.2.3 Формат заголовка сообщений для протокола маршрутизации OSPF

Поле версия определяет версию протокола (= 2). Поле тип идентифицирует функцию сообщения согласно таблице 4.2.11.2.2:

Таблица 4.2.11.2.2. Коды поля тип

Тип	Значение
1	Hello (используется для проверки доступности маршрутизатора).
2	Описание базы данных (топология).
3	Запрос состояния канала.
4	Изменение состояния канала.
5	Подтверждение получения сообщения о статусе канала.

Поле длина пакета определяет длину блока в октетах, включая заголовок. Идентификатор области - 32-битный код, идентифицирующий область, которой данный пакет принадлежит. Все OSPF-пакеты ассоциируются с той или иной областью. Большинство из них не преодолевает более одного шага. Пакеты, путешествующие по виртуальным каналам, помечаются идентификатором опорной области (backbone) 0.0.0.0. Поле контрольная сумма содержит контрольную сумму IP-пакета, включая поле типа идентификации. Контрольное суммирование производится по модулю 1. Поле тип идентификации может принимать значения 0 при отсутствии контроля доступа, и 1 при наличии контроля. В дальнейшем функции поля будут расширены. Важную функцию в OSPF-сообщениях выполняет одно-октетное поле опции, оно присутствует в сообщениях типа Hello, объявление состояния канала и описание базы данных. Особую роль в этом поле играют младшие биты E и T:

Формат поля *опции*

Бит E характеризует возможность внешней маршрутизации и имеет значение только в сообщениях типа Hello, в остальных сообщениях этот бит должен быть обнулен. Если E=0, то данный маршрутизатор не будет посылать или принимать маршрутную информацию от внешних автономных систем. Бит T определяет сервисные возможности маршрутизатора (TOS). Если T=0, это означает, что маршрутизатор поддерживает только один вид услуг (TOS=0) и он не пригоден для маршрутизации с учетом вида услуг. Такие маршрутизаторы, как правило, не используются для транзитного трафика.

Протокол OSPF использует сообщение типа Hello для взаимообмена данными между соседними маршрутизаторами. Структура пакетов этого типа показана на рис. 4.2.11.2.4.

Рис. 4.2.11.2.4 Формат сообщения Hello в протоколе OSPF

Поле сетевая маска соответствует маске субсети данного интерфейса. Например, если интерфейс принадлежит сети класса B и третий байт служит для выделения нужной субсети, то сетевая маска будет иметь вид 0xFFFFFF00.

Поле время между Hello содержит значение времени в секундах, между сообщениями Hello. Поле опции характеризует возможности, которые предоставляет данный маршрутизатор. Поле приоритет характеризует уровень приоритета маршрутизатора (целое положительное число), используется при выборе резервного (backup) маршрутизатора. Если приоритет равен нулю, данный маршрутизатор никогда не будет использован в качестве резервного. Поле время отключения маршрутизатора определяет временной интервал в секундах, по истечении которого "молчащий" маршрутизатор считается вышедшим из строя. IP-адреса маршрутизаторов, записанные в последующих полях, указывают место, куда следует послать данное сообщение. Поля IP-адрес соседа k образуют список адресов соседних маршрутизаторов, откуда за последнее время были получены сообщения Hello.

Маршрутизаторы обмениваются сообщениями из баз данных OSPF, чтобы инициализировать, а в дальнейшем актуализовать свои базы данных, характеризующие топологию сети. Обмен происходит в режиме клиент-сервер. Клиент подтверждает получение каждого сообщения. Формат пересылки записей из базы данных представлен на рис. 4.2.11.2.5.

Рис. 4.2.11.2.5 Формат OSPF-сообщений о маршрутах

Поля, начиная с тип канала, повторяются для каждого описания канала. Так как размер базы данных может быть велик, ее содержимое может пересылаться по частям. Для реализации этого используются биты I и M. Бит I устанавливается в 1 в стартовом сообщении, а бит M принимает единичное состояние для сообщения, которые являются продолжением. Бит S определяет то, кем послано сообщение (S=1 для сервера, S=0 для клиента, этот бит иногда

имеет имя MS). Поле номер сообщения по порядку служит для контроля пропущенных блоков. Первое сообщение содержит в этом поле случайное целое число M, последующие M+1, M+2,...M+L. Поле тип канала может принимать следующие значения:

Таблица 4.2.11.2.3. Коды типов состояния каналов (LS)

LS-тип	Описание объявления о маршруте
1	Описание каналов маршрутизатора, то есть состояния его интерфейсов.
2	Описание сетевых каналов. Это перечень маршрутизаторов, непосредственно связанных с сетью.
3 или 4	Сводное описание каналов, куда входят маршруты между отдельными областями сети. Эта информация поступает от пограничных маршрутизаторов этих зон. Тип 3 приписан маршрутам, ведущим к сетям, а тип 4 характеризует маршруты, ведущие к пограничным маршрутизаторам автономной системы.
5	Описания внешних связей автономной системы. Такие маршруты начинаются в пограничных маршрутизаторах AS.

Поле идентификатор канала определяет его характер, в зависимости от этого идентификатором может быть IP-адрес маршрутизатора или сети. Маршрутизатор, рекламирующий канал определяет адрес этого маршрутизатора. Поле порядковый номер канала позволяет маршрутизатору контролировать порядок прихода сообщений и их потерю. Поле возраст канала определяет время в секундах с момента установления связи. После обмена сообщениями с соседями маршрутизатор может выяснить, что часть данных в его базе устарела. Он может послать своим соседям запрос с целью получения свежей маршрутной информации о каком-то конкретном канале связи. Сосед, получивший запрос, высылает необходимую информацию. Запрос посылается в соответствии с форматом, показанном ниже (рис. 4.2.11.2.6):

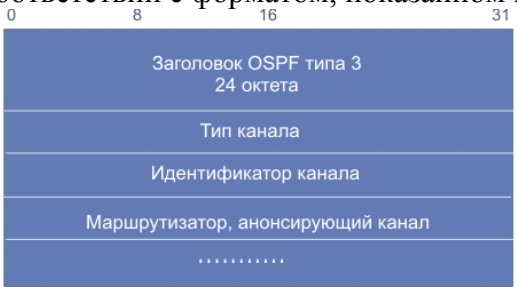


Рис. 4.2.11.2.6 Формат OSPF-запроса маршрутной информации

Три поля этого запроса повторяются согласно числу каналов, информация о которых запрашивается. Если список достаточно длинен, может потребоваться несколько запросов. Маршрутизаторы посылают широковещательные (или мультикастинговые) сообщения об изменении состояния своих непосредственных связей. Такое сообщение содержит список объявлений, имеющих формат (рис. 4.2.11.2.7):

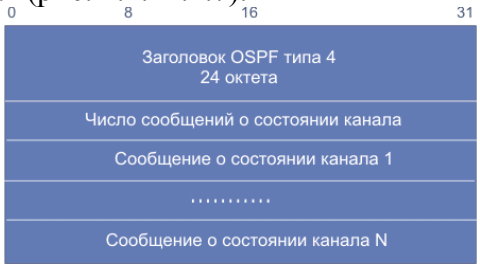


Рис. 4.2.11.2.7 Сообщение об изменении маршрутов

Сообщения об изменениях маршрутов могут быть вызваны следующими причинами:

1. Возраст маршрута достиг предельного значения (lsrefreshtime).
2. Изменилось состояние интерфейса.
3. Произошли изменения в маршрутизаторе сети.

4. Произошло изменение состояния одного из соседних маршрутизаторов.
5. Изменилось состояние одного из внутренних маршрутов (появление нового, исчезновение старого и т.д.)
6. Изменение состояния межзонного маршрута.
7. Появление нового маршрутизатора, подключенного к сети.
8. Вариация виртуального маршрута одним из маршрутизаторов.
9. Возникли изменения одного из внешних маршрутов.
10. Маршрутизатор перестал быть пограничным для данной as (например, перезагрузился).
Каждое сообщение о состоянии канала начинается с заголовка - "объявление состояния канала" (LS- link state). Формат этого типа заголовка приведен ниже (20 октетов):

Рис. 4.2.11.2.8 Формат OSPF-сообщения, описывающего состояние канала

Поле возраст ls информации (рис. 4.2.11.2.8) определяет время в секундах с момента объявления состояния канала. Поле опции содержит значения типов сервиса (TOS), поддерживаемые маршрутизатором, рассылающим маршрутную информацию. Поле тип LS (тип состояния канала) может принимать значения, описанные выше в табл. 4.2.11.2.3. Следует обратить внимание, что код, содержащийся в этом поле, определяет формат сообщения. Поле длина задает размер сообщения в октетах, включая заголовок. В результате получается сообщение с форматом, показанным на рис. 4.2.11.2.9. Зарезервированный октет должен быть обнулен. Идентификатор связи определяет тип маршрутизатора, подключенного к каналу. Действительное значение этого поля зависит от поля тип. В свою очередь информация о канале также зависит от поля тип. Число tos определяет многообразие метрик, соответствующих видам сервиса, для данного канала. Последовательность описания метрик задается величиной кода TOS. Таблица кодов TOS, принятых в OSPF протоколе приведена ниже.

Таблица 4.2.11.2.4. Коды типа сервиса (TOS)

OSPF-код	**TOS-коды**	**TOS(RFC-1349)**
0	0000	Обычный сервис
2	0001	Минимизация денежной стоимости
4	0010	Максимальная надежность
8	0100	Максимальная пропускная способность
16	1000	Минимальная задержка

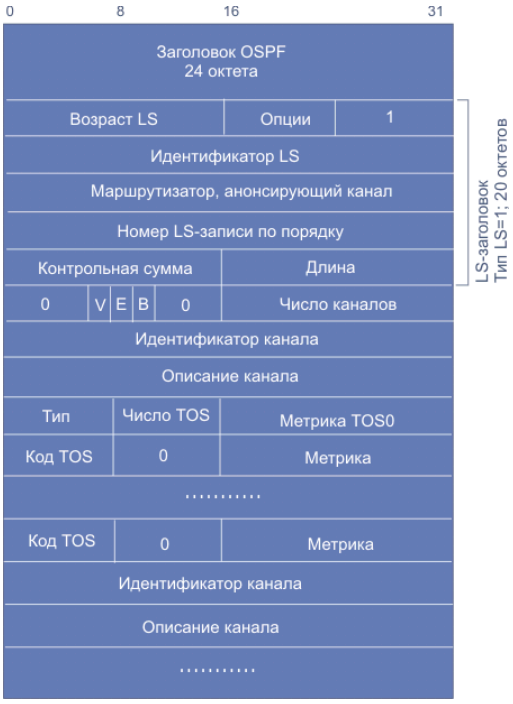


Рис. 4.2.11.2.9 Формат описания типа канала с LS=1

Если бит V=1 (virtual), маршрутизатор является оконечной точкой активного виртуального канала. Если бит E (external) равен 1, маршрутизатор является пограничным для автономной системы. Бит B=1 (border) указывает на то, что маршрутизатор является пограничным для данной области. Поле тип может принимать значения, приведенные в таблице 4.2.11.2.5.

Таблица 4.2.11.2.5. Коды типов связей (см. рис. 4.2.11.2.9)

Код типа связи	Описание
1	Связь с другим маршрутизатором по схеме точка-точка
2	Связь с транзитной сетью
3	Связь с оконечной сетью
4	Виртуальная связь (например, опорная сеть или туннель)

Поле идентификатор канала характеризует объект, с которым связывается маршрутизатор. Примеры идентификаторов представлены в таблице:

Таблица 4.2.11.2.6. Коды идентификаторов канала

Код идентификатора	**Описание**
1	Идентификатор соседнего маршрутизатора
2	IP-адрес основного маршрутизатора (по умолчанию)
3	IP-адрес сети/субсети
4	Идентификатор соседнего маршрутизатора

Маршрутизатор, получивший OSPF-пакет, посылает подтверждение его приема. Этот вид пакетов имеет тип=5 и структуру, отображенную на рис. 4.2.11.2.10. Получение нескольких объявлений о состоянии линий может быть подтверждено одним пакетом. Адресом места назначения этого пакета может быть индивидуальный маршрутизатор, группа маршрутизаторов или все маршрутизаторы автономной системы.

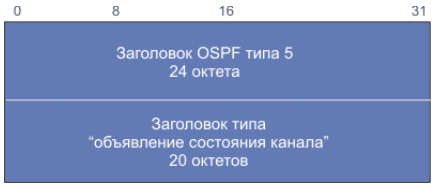


Рис. 4.2.11.2.10 Формат сообщения о получении OSPF-пакета

Рекламирование сетевых связей относится к типу 2. Сообщения посылаются для каждой транзитной сети в автономной системе. Транзитной считается сеть, которая имеет более одного маршрутизатора. Сообщение о сетевых связях должно содержать информацию обо всех маршрутизаторах, подключенных к сети, включая тот, который рассылает эту информацию. Расстояние от сети до любого подключенного маршрутизатора равно нулю для всех видов сервиса (TOS), поэтому поля TOS и метрики в этих сообщениях отсутствуют. Формат сообщения о транзитных сетевых связях показан на рис. 4.2.11.2.11.

Следует помнить, что приведенные ниже сообщения должны быть снабжены стандартными 24-октетными OSPF-заголовками (на рис. 4.2.11.2.11 отсутствует).

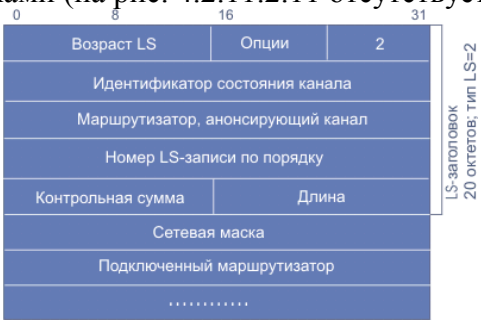


Рис. 4.2.11.2.11 Формат сообщения о сетевых связях (тип LS=2)

Сетевая маска относится к описываемой сети, а поле подключенный маршрутизатор содержит идентификатор маршрутизатора, работающего в сети. Информация об адресатах в пределах автономной системы передается LS-сообщениями типа 3 и 4. Тип 3 работает для IP-сетей. В этом случае в качестве идентификатора состояния канала используется IP-адрес сети. Если же адресатом является пограничный маршрутизатор данной AS, то используется LS-сообщение типа 4, а в поле идентификатор состояния канала записывается OSPF-идентификатор этого маршрутизатора. Во всех остальных отношениях сообщения 3 и 4 имеют идентичные форматы (рис. 4.2.11.2.12):

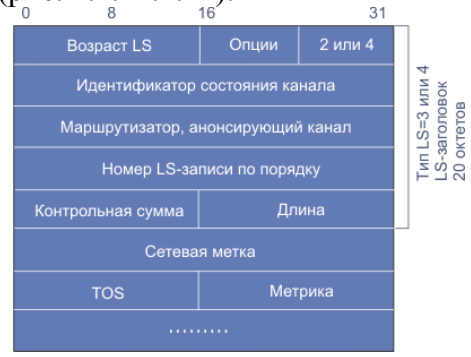


Рис. 4.2.11.2.12 Формат сообщений об адресатах в пределах автономной системы

Поля, следующие после заголовка, повторяются в соответствии с числом описываемых объектов. Рекламирование внешних маршрутов относится к пятому типу. Эта информация рассылается пограничными маршрутизаторами. Информация о каждом внешнем адресате, известном маршрутизатору, посылается независимо. Этот вид описания используется и для маршрутов по умолчанию, для которых идентификатор состояния канала устанавливается равным 0.0.0.0 (аналогичное значение принимает при этом и сетевая маска). Формат такого сообщения представлен на рис. 4.2.11.2.13.

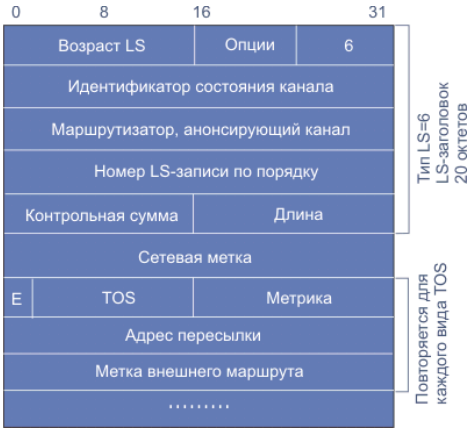


Рис. 4.2.11.2.13 Формат описания внешних маршрутов

Сетевая маска характеризует место назначения рекламируемого маршрута. Так для сети класса А маска может иметь вид 0xFF000000. Последующий набор полей повторяется для каждого вида TOS. Поля для TOS=0 заполняются всегда, и это описание является первым. Бит Е характеризует внешнюю метрику. Если Е=0, то она может непосредственно (без преобразования) сравниваться с метриками других каналов. При Е=1 метрика считается больше любой метрики. Поле адрес пересылки указывает на место, куда будут пересылаться данные, адресованные рекламируемым маршрутом. Если адрес пересылки равен 0.0.0.0, данные посылаются пограничному маршрутизатору автономной системы - источнику данного сообщения. Метка внешнего маршрута - 32-битовое число, присваиваемое каждому внешнему маршруту. Эта метка самим протоколом OSPF не используется и предназначена для информирования других автономных систем при работе внешних протоколов маршрутизации. Маршрутная таблица OSPF содержит в себе:

- IP-адрес места назначения и маску;
- тип места назначения (сеть, граничный маршрутизатор и т.д.);
- тип функции (возможен набор маршрутизаторов для каждой из функций TOS);
- область (описывает область, связь с которой ведет к цели, возможно несколько записей данного типа, если области действия граничных маршрутизаторов перекрываются);
- тип пути (характеризует путь как внутренний, межобластной или внешний, ведущий к AS);
- цена маршрута до цели;
- очередной маршрутизатор, куда следует послать дейтограмму;
- объявляющий маршрутизатор (используется для межобластных обменов и для связей автономных систем друг с другом).

Преимущества OSPF:

1. Для каждого адреса может быть несколько маршрутных таблиц, по одной на каждый вид IP-операции (TOS).
2. Каждому интерфейсу присваивается безразмерная цена, учитывающая пропускную способность, время транспортировки сообщения. Для каждой IP-операции может быть присвоена своя цена (коэффициент качества).
3. При существовании эквивалентных маршрутов OSFP распределяет поток равномерно по этим маршрутам.
4. Поддерживается адресация субсетей (разные маски для разных маршрутов).
5. При связи точка-точка не требуется IP-адрес для каждого из концов. (Экономия адресов!)
6. Применение мультикастинга вместо широковещательных сообщений снижает загрузку не вовлеченных сегментов.

Недостатки:

1. Трудно получить информацию о предпочтительности каналов для узлов, поддерживающих другие протоколы, или со статической маршрутизацией.
2. OSPF является лишь внутренним протоколом.

18. Маршрутизация: маршрутизация второго уровня.

См. 7 вопрос. А лучше лекции.

19. Понятие маски подсети, ее назначение. Безклассовая модель представления сетевых адресов.

Важным элементом разбиения адресного пространства Internet являются подсети. Подсеть - это подмножество сети, не пересекающееся с другими подсетями. Это означает, что сеть организации (скажем, сеть класса С) может быть разбита на фрагменты, каждый из которых будет составлять подсеть. Реально, каждая подсеть соответствует физической локальной сети (например, сегменту Ethernet). Вообще говоря, подсети придуманы для того, чтобы обойти ограничения физических сетей на число узлов в них и максимальную длину кабеля в сегменте сети. Например, сегмент тонкого Ethernet имеет максимальную длину 185 м и может включать до 32 узлов. Самая маленькая сеть - класса С - может состоять из 254 узлов. Для того чтобы достичь этой цифры, надо объединить несколько физических сегментов сети. Сделать это можно либо с помощью физических устройств (например, репитеров), либо при помощи машин-шлюзов. В первом случае разбиения на подсети не требуется, т.к. логически сеть выглядит как одно целое. При использовании шлюза сеть разбивается на подсети (Рисунок 12).

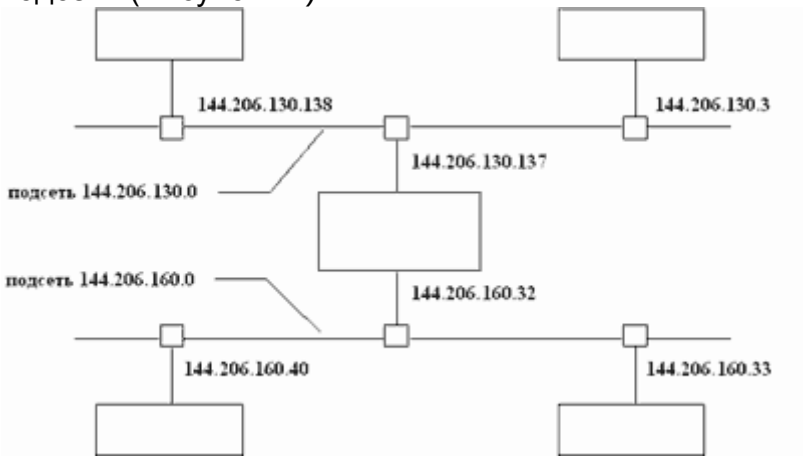


Рисунок 12 Схема разбиения адресного пространства сети на подсети

На Рисунок 12 изображен фрагмент сети класса В - 144.206.0.0, состоящий из двух подсетей - 144.206.130.0 и 144.206.160.0. В центре схемы изображена машина шлюз, которая связывает подсети. Эта машина имеет два сетевых интерфейса и, соответственно, два IP-адреса.

В принципе, разбивать сеть на подсети необязательно. Можно использовать адреса сетей другого класса (с меньшим максимальным количеством узлов). Но при этом возникает, как минимум, два неудобства:

В сети, состоящей из одного сегмента Ethernet, весь адресный пул сети не будет использован, т.к., например, для сети класса С (самой маленькой с точки зрения количества узлов в ней), из 254 возможных адресов можно использовать только 32;

Все машины за пределами организации, которым разрешен доступ к компьютерам сети данной организации, должны знать шлюзы для каждой из сетей. Структура сети становится открытой во внешний мир. Любые изменения структуры могут вызвать ошибки маршрутизации. При использовании подсетей внешним машинам надо знать только шлюз всей сети организации. Маршрутизация внутри сети - это ее внутреннее дело.

Разбиение сети на подсети использует ту часть IP-адреса, которая закреплена за номерами хостов. Администратор сети может замаскировать часть IP-адреса и использовать ее для назначения номеров подсетей. Фактически, способ разбиения адреса на две части, теперь будет применятся к адресу хоста из IP-адреса сети, в которой организуется разбиение на подсети.

Маска подсети - это четыре байта, которые накладываются на IP-адрес для получения номера подсети. Например, маска 255.255.255.0 позволяет разбить сеть класса B на 254 подсети по 254 узла в каждой. На Рисунок 13 приведено маскирование подсети 144.206.160.0 из предыдущего примера.

На приведенной схеме (Рисунок 13) сеть класса B (номер начинается с 10) разбивается на подсети маской 255.255.224.0. При этом первые два байта задают адрес сети и не участвуют в разбиении на подсети. Номер подсети задается тремя старшими битами третьего байта маски. Такая маска позволяет получить 6 подсетей. Для нумерации подсети нельзя использовать номер 000 и номер 111. Номер 160 задает 5-ю подсеть в сети 144.206.0.0. Для нумерования машин в подсети можно использовать оставшиеся после маскирования 13 битов, что позволяет создать подсеть из 8190 узлов. Честно говоря, в настоящее время такой сети в природе не существует и РНЦ "Курчатовский Институт", которому принадлежит сеть 144.206.0.0, рассматривает возможность пересмотра маски подсетей. Перестроить сеть, состоящую из более чем 400 машин, не такая простая задача, так как ей управляет 4 администратора, которые должны изменить маски на всех машинах сети. Ряд компьютеров работает в круглосуточном режиме и все изменения надо произвести в тот момент, когда это минимально скажется на работе пользователей сети. Данный пример показывает насколько внимательно следует подходить к вопросам планирования архитектуры сети и ее разбиения на подсети. Многие проблемы можно решить за счет аппаратных средств построения сети.

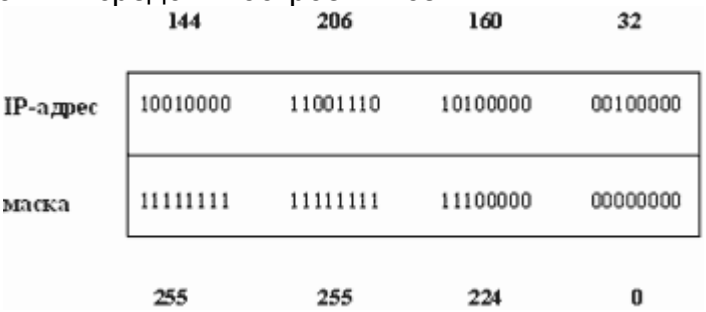


Рисунок 13 Схема маскирования и вычисления номера подсети

К сожалению, подсети не только решают, но также и создают ряд проблем. Например, происходит потеря адресов, но уже не по причине физических ограничений, а по причине принципа построения адресов подсети. Как было видно из примера, выделение трех битов на адрес подсети не приводит к образованию 8-ми подсетей. Подсетей образуется только 6, так как номера сетей 0 и 7 использовать в силу специального значения IP-адресов, состоящих из 0 и единиц, нельзя. Таким образом, все комбинации адресов хоста внутри подсети, которые можно было бы связать с этими номерами, придется забыть. Чем шире маска подсети (чем больше

места отводится на адрес хоста), тем больше потерь. В ряде случаев приходится выбирать между приобретением еще одной сети или изменением маски. При этом физические ограничения могут быть превышены за счет репитеров, хабов и т. п.

Бесклассовая адресация (англ. *Classless InterDomain Routing*, англ. *CIDR*) — метод IP-адресации, позволяющий гибко управлять пространством IP-адресов, не используя жёсткие рамки классовой адресации. Использование этого метода позволяет экономно использовать конечный ресурс IP-адресов.

Бесклассовая адресация основывается на *переменной длине* маски подсети (англ. *Variable Lenght Subnet Mask* — VLSM), в то время, как в классовой адресации длина маски строго фиксирована 0,1,2 или 3 установленными байтами. Вот пример записи IP-адреса с применением бесклассовой адресации: **10.1.2.33/27**.

октеты IP-адреса	10								1								2								33							
биты IP-адреса	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1
биты маски подсети	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
октеты маски подсети	255								255								255								224							

В данном примере видно, что в маске подсети 27 бит слева выставлены в единицу (так называемые *значащие биты*. В таком случае говорят о длине маски подсети в 27 бит (/27 — на сленге "**слэш двадцать семь**").

Вот ещё один пример записи адреса в бесклассовой нотации: **172.16.0.1/12**.

октеты IP-адреса	172								16								0								1							
биты IP-адреса	1	0	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
биты маски подсети	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
октеты маски подсети	255								240								0								0							

Для удобства записи IP-адрес в модели CIDR часто представляется в виде a.b.c.d / n, где a.b.c.d — IP адрес, n — количество бит в сетевой части.

Пример: 137.158.128.0/17.

Маска сети для этого адреса: 17 единиц (сетевая часть), за ними 15 нулей (хостовая часть), что в октетном представлении равно

11111111.11111111.10000000.00000000 = 255.255.128.0.

Представив IP-адрес в двоичном виде и побитно умножив его на маску сети, мы получим номер сети (все нули в хостовой части). Номер хоста в этой сети мы можем получить, побитно умножив IP-адрес на инвертированную маску сети.

Пример: IP = 205.37.193.134/26 или, что то же,

IP = 205.37.193.134 netmask = 255.255.255.192.

Распишем в двоичном виде:

IP = 11001101 00100101 11000111 10000110

маска = 11111111 11111111 11111111 11000000

Умножив побитно, получаем номер сети (в хостовой части - нули):

network = 11001101 00100101 11000111 10000000

или, в октетном представлении, 205.37.193.128/26, или, что то же, 205.37.193.128 netmask 255.255.255.192.

Хостовая часть рассматриваемого IP адреса равна 000110, или 6. Таким образом, 205.37.193.134/26 адресует хост номер 6 в сети 205.37.193.128/26. В классовой модели адрес 205.37.193.134 определял бы хост 134 в сети класса C 205.37.193.0, однако указание маски сети (или количества бит в сетевой части) однозначно определяет принадлежность адреса к бесклассовой модели.

Очевидно, что сети классов A, B, C в бесклассовой модели представляются при помощи масок, соответственно, 255.0.0.0 (или /8), 255.255.0.0 (или /16) и 255.255.255.0 (или /24).

20. Маршрутизация: маршрутизация третьего уровня.

См. 7 вопрос. А лучше лекции.

21. Разрешение сетевых имен с помощью DNS. Протокол ARP.

Имя упрощает обращение к узлу, поскольку его легче запомнить, чем IP-адрес. Имена узлов используются практически везде, где есть TCP/IP.

Имя узла — это псевдоним, назначенный администратором компьютеру для идентификации узла, поддерживающего TCP/IP. Имя узла иногда не совпадает с NetBIOS-именем данного компьютера и содержит до 256 символов. Одному узлу можно назначить несколько имен.

***Разрешение имени**, узла* (host name resolution) — это процесс определения соответствующего ему IP-адреса.

Доменная система имен (DNS) это часть семейства протоколов и утилит TCP/IP. Microsoft и другие компании предлагают различные версии DNS, работающие на разнообразных операционных системах (в основном на вариантах Unix). Слово domain в названии DNS относится к доменам в Internet, а не к доменной модели NT.

1. Когда пользователь вводит команду, применяя FQDN или другое имя узла, сервер DNS ищет это имя в базе данных и разрешает его в IP-адрес.

Если сервер DNS не отвечает на запрос, то с интервалами 5, 10, 20, 40, 5, 10 и 20 секунд выполняются повторные попытки. Если сервер DNS не отвечает и на эти запросы, а другие методы, например, сервер имен NetBIOS или файл LMHOSTS недоступны, то процесс прекращается и генерируется сообщение об ошибке.

2. После того как имя узла разрешено, по протоколу ARP определяется адрес сетевого адаптера. Если узел назначения находится в локальной сети, то это реализуется при помощи кэша ARP или широковещания. Если же узел получатель находится в

удаленной сети, то ARP получает адрес маршрутизатора, который может перенаправить запрос.

Internet подразделяется на домены, каждый из которых обслуживает различные группы пользователей. К таковым доменам относятся домены .com, .edu, .gov и .mil. Ими управляет Internet-сервер первого уровня, получивший название корневого сервера имен (это название становится понятным, если представлять себе Internet как древовидную структуру).

Система именования доменов Internet сначала обращается к Internet-серверам первого уровня, а затем "спускается" по дереву серверов. Когда вы набираете адрес, ваш локальный сервер DNS просматривает свою базу данных и кэширует требуемую информацию. Если локальный сервер не содержит IP-адрес, он передает запрос корневому серверу имен. После чего корневой сервер имен возвращает вашему серверу DNS адрес соответствующего сервера имен. В свою очередь ваш сервер DNS обращается с запросом к серверу имен в поисках адреса сервера на следующем уровне и далее процесс повторяется.

Например, если вы хотите обратится на узел http://www.winntmag.com, ваш сервер DNS обращается к серверу домена .com в поисках адреса сервера имен winntmag в домене .com. Локальный сервер DNS использует адрес, полученный по этому запросу, для обращения к серверу winntmag.com в поисках адреса хоста Web-узла.

Приведенное выше описание применимо к последовательным (итерационным) запросам, которые DNS выполняет от сервера к серверу. DNS также может выполнять рекурсивный запрос, при котором сервер имен доменов передает результаты поиска непосредственно исходному клиенту.

Чтобы сделать оба этих поиска более эффективными, сервер DNS кэширует ответы в каждой точке поиска. Если после связи с узлом http://www.winntmag.com вы захотите обратится к другому серверу .com, ваш сервер DNS уже будет знать адрес сервера домена .com. Если же вы захотите связаться с другим компьютером в домене winntmag.com, ваш сервер DNS уже содержит в кэш-памяти адрес сервера имен winntmag.com, и нет необходимости запрашивать его еще раз.

Протокол ARP

Протокол ARP (Address Resolution Protocol, Протокол распознавания адреса) предназначен для преобразования IP-адресов в MAC-адреса, часто называемые также физическими адресами.

MAC расшифровывается как Media Access Control, контроль доступа к среде передачи. MAC-адреса идентифицируют устройства, подключенные к физическому каналу, пример MAC-адреса - адрес Ethernet.

Для передачи IP-дейтаграммы по физическому каналу (будем рассматривать Ethernet) требуется инкапсулировать эту дейтаграмму в кадр Ethernet и в заголовке кадра указать адрес Ethernet-карты, на которую будет доставлена эта дейтаграмма для ее последующей обработки вышестоящим по стеку протоколом IP. IP-адрес, включенный в заголовок дейтаграммы, адресует IP-интерфейс какого-либо узла сети и не заключает в себе никаких указаний ни на физическую среду передачи, к которой подключен этот интерфейс, ни тем более на физический адрес устройства (если таковой имеется), с помощью которого этот интерфейс сообщается со средой.

Поиск по данному IP-адресу соответствующего Ethernet-адреса производится протоколом ARP, функционирующим на уровне доступа к среде передачи. Протокол поддерживает в оперативной памяти динамическую arp-таблицу в целях кэширования полученной информации. Порядок функционирования протокола следующий.

С межсетевого уровня поступает IP-дейтаграмма для передачи в физический канал (Ethernet), вместе с дейтаграммой передается, среди прочих параметров, IP-адрес узла назначения. Если в arp-таблице не содержится записи об Ethernet-адресе, соответствующем нужному IP-адресу, модуль arp ставит дейтаграмму в очередь и формирует широковещательный запрос. Запрос получают все узлы, подключенные к данной сети; узел, опознавший свой IP-адрес, отправляет arp-ответ (arp-response) со значением своего адреса Ethernet. Полученные данные заносятся в таблицу, ждущая дейтаграмма извлекается из очереди и передается на инкапсуляцию в кадр Ethernet для последующей отправки по физическому каналу.
ARP-запрос или ответ включается в кадр Ethernet непосредственно после заголовка кадра.

ARP для дейтаграмм, направленных в другую сеть

Дейтаграмма, направленная во внешнюю (в другую) сеть, должна быть передана маршрутизатору. Предположим, хост A отправляет дейтаграмму хосту B через маршрутизатор G. Несмотря на то, что в заголовке дейтаграммы, отправляемой из A, в поле “Destination” указан IP-адрес B, кадр Ethernet, содержащий эту дейтаграмму, должен быть доставлен маршрутизатору. Это достигается тем, что IP-модуль при вызове ARP-модуля передает тому вместе с дейтаграммой в качестве IP-адреса узла назначения адрес маршрутизатора, извлеченный из таблицы маршрутов. Таким образом, дейтаграмма с адресом B инкапсулируется в кадр с MAC-адресом G:

IP Source: A IP Destination: B	Заголовок дейтаграммы
Ethernet Source: A Ethernet Destination: G	Заголовок кадра Ethernet

Модуль Ethernet на маршрутизаторе G получает из сети этот кадр, так как кадр адресован ему, извлекает из кадра данные (то есть дейтаграмму) и отправляет их для обработки модулю IP. Модуль IP обнаруживает, что дейтаграмма адресована не ему, а хосту B, и по своей таблице маршрутов определяет, куда ее следует переслать. Далее дейтаграмма опять опускается на нижний уровень, к соответствующему физическому интерфейсу, которому передается в качестве IP-адреса узла назначения адрес следующего маршрутизатора, извлеченный из таблицы маршрутов, или сразу адрес хоста B, если маршрутизатор G может доставить дейтаграмму непосредственно к нему.

Proxy ARP

ARP-ответ может отправляться не обязательно искомым узлом, вместо него это может сделать другой узел. Такой механизм называется *proxy ARP*.
Рассмотрим пример (рис. 2.6.1). Удаленный хост A подключается по коммутируемой линии к сети 194.84.124.0/24 через сервер доступа G. Сеть 194.84.124.0 на физическом уровне представляет собой Ethernet. Сервер G выдает хосту A IP-адрес 194.84.124.30, принадлежащий сети 194.84.124.0. Следовательно, любой узел этой сети, например, хост B, полагает, что может непосредственно отправить дейтаграмму хосту A, поскольку они находятся в одной IP-сети.

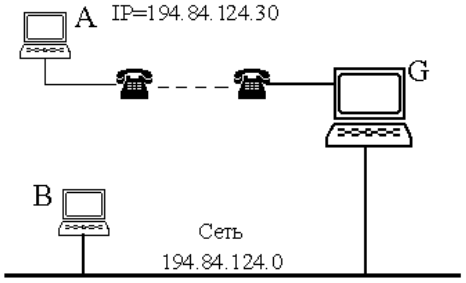


Рис. 2.6.1. Proxy ARP

IP-модуль хоста B вызывает ARP-модуль для определения физического адреса A. Однако вместо A (который, разумеется, откликнуться не может, потому что физически не подключен к сети Ethernet) откликается сервер G, который и возвращает свой Ethernet-адрес как физический адрес хоста A. Вслед за этим B отправляет, а G получает кадр, содержащий дейтаграмму для A, которую G отправляет адресату по коммутируемому каналу.

22. Понятие фреймов Ethernet (IEEE 802.3 Packet Framing), изменения в Ethernet II.

В 1980 году фирмы DEC, Intel и Xerox совместно разработали и опубликовали стандарт Ethernet версии II для сети, построенной на основе коаксиального кабеля, который стал последней версией фирменного стандарта Ethernet. Поэтому фирменную версию стандарта Ethernet называют стандартом Ethernet DIX или Ethernet П.

На основе стандарта Ethernet DIX был разработан стандарт IEEE 802.3, который во многом совпадает со своим предшественником, но некоторые различия все же имеются. В то время как в стандарте IEEE 802.3 различаются уровни MAC и LLC, в оригинальном Ethernet оба эти уровня объединены в единый канальный уровень. В Ethernet DIX определяется протокол тестирования конфигурации (Ethernet Configuration Test Protocol), который отсутствует в IEEE 802.3. Несколько отличается и формат кадра, хотя минимальные и максимальные размеры кадров в этих стандартах совпадают. Часто для того, чтобы отличить Ethernet, определенный стандартом IEEE, и фирменный Ethernet DIX, первый называют технологией 802.3, а за фирменным оставляют название Ethernet без дополнительных обозначений.

Для передачи двоичной информации по кабелю для всех вариантов физического уровня технологии Ethernet, обеспечивающих пропускную способность 10 Мбит/с, используется манчестерский код.

Все виды стандартов Ethernet (в том числе Fast Ethernet и Gigabit Ethernet) используют один и тот же метод разделения среды передачи данных — метод CSMA/CD.

Этот метод применяется исключительно в сетях с логической общей шиной (к которым относятся и радиосети, породившие этот метод). Все компьютеры такой сети имеют непосредственный доступ к общей шине, поэтому она может быть использована для передачи данных между любыми двумя узлами сети. Одновременно все компьютеры сети имеют возможность немедленно (с учетом задержки распространения сигнала по физической среде) получить данные, которые любой из компьютеров начал передавать на общую шину (рис. 3.3). Простота схемы подключения — это один из факторов, определивших успех стандарта Ethernet. Говорят, что кабель, к которому подключены все станции, работает в режиме коллективного доступа (Multiply Access, MA).

Рис. 3.3. Метод случайного доступа CSMA/CD

Все данные, передаваемые по сети, помещаются в кадры определенной структуры и снабжаются уникальным адресом станции назначения.

Чтобы получить возможность передавать кадр, станция должна убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая также называется несущей частотой (carrier-sense, CS). Признаком незанятости среды является отсутствие на ней несущей частоты (5-10 МГц).Если среда свободна, то узел имеет право начать передачу кадра.

Кадр данных всегда сопровождается *преамбулой (preamble),* которая состоит из 7 байт, состоящих из значений 10101010, и 8-го байта, равного 10101011. Преамбула нужна для вхождения приемника в побитовый и побайтовый синхронизм с передатчиком. Все станции, подключенные к кабелю, могут распознать факт передачи кадра, и та станция, которая узнает собственный адрес в заголовках кадра, записывает его содержимое в свой внутренний буфер, обрабатывает полученные данные, передает их вверх по своему стеку, а затем посылает по кабелю кадр-ответ.

Коллизия — это нормальная ситуация в работе сетей Ethernet. Коллизия часто возникает из-за того, что один узел начинает передачу раньше другого, но до второго узла сигналы первого просто не успевают дойти к тому времени, когда второй узел решает начать передачу своего кадра. Все станции одновременно наблюдают за возникающими на кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется *обнаружение коллизии.*

Обнаружившая коллизию передающая станция обязана прекратить передачу и сделать паузу в течение короткого случайного интервала времени. Затем она может снова предпринять попытку захвата среды и передачи кадра. Таким образом, случайная пауза может принимать значения от 0 до 52,4 мс. Если 16 последовательных попыток передачи кадра вызывают коллизию, то передатчик должен прекратить попытки и отбросить этот кадр. Из описания метода доступа видно, что он носит вероятностный характер, и вероятность успешного получения в свое распоряжение общей среды зависит от загруженности сети, то есть от интенсивности возникновения в станциях потребности в передаче кадров.

Так как в худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга станциями сети (в одну сторону проходит неискаженный сигнал, а на обратном пути распространяется уже искаженный коллизией сигнал), то это время называется *временем двойного оборота (Path Delay Value, PDV).*

В стандарте Ethernet принято, что минимальная длина поля данных кадра составляет 46 байт (что вместе со служебными полями дает минимальную длину кадра 64 байт, а вместе с преамбулой — 72 байт или 576 бит). Отсюда может быть определено ограничение ***на расстояние между станциями***: в 10-мегабитном Ethernet время передачи кадра минимальной длины равно 575 битовых интервалов, следовательно, время двойного оборота должно быть меньше 57,5 мкс. Расстояние, которое сигнал может пройти за это время, зависит от типа кабеля и для толстого коаксиального кабеля равно примерно 13 280 м. Учитывая, что за это время сигнал

должен пройти по линии связи дважды, расстояние между двумя узлами не должно быть больше 6 635 м. В стандарте величина этого расстояния выбрана существенно меньше, с учетом других, более строгих ограничений.

Спецификации физической среды Ethernet

Исторически первые сети технологии Ethernet были созданы на коаксиальном кабеле диаметром 0,5 дюйма. В дальнейшем были определены и другие спецификации физического уровня для стандарта Ethernet, позволяющие использовать различные среды передачи данных. Метод доступа CSMA/CD и все временные параметры остаются одними и теми же для любой спецификации физической среды технологии Ethernet 10 Мбит/с.

Физические спецификации технологии Ethernet на сегодняшний день включают следующие среды передачи данных.

- 10Base-5 — коаксиальный кабель диаметром 0,5 дюйма, называемый «толстым» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента — 500 метров (без повторителей).
- 10Base-2 — коаксиальный кабель диаметром 0,25 дюйма, называемый «тонким» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента — 185 метров (без повторителей).
- 10Base-T — кабель на основе неэкранированной витой пары (Unshielded Twisted Pair, UTP). Образует звездообразную топологию на основе концентратора. Расстояние между концентратором и конечным узлом — не более 100 м.
- 10Base-F — волоконно-оптический кабель. Топология аналогична топологии стандарта 10Base-T. Имеется несколько вариантов этой спецификации — FOIRL (расстояние до 1000 м), 10Base-FL (расстояние до 2000 м), 10Base-FB (расстояние до 2000 м).

Число 10 в указанных выше названиях обозначает битовую скорость передачи данных этих стандартов — 10 Мбит/с, а слово Base — метод передачи на одной базовой частоте 10 МГц (в отличие от методов, использующих несколько несущих частот, которые называются Broadband — широкополосными). Последний символ в названии стандарта физического уровня обозначает тип кабеля

23. Протоколы маршрутизации: RIP, OSPF, BGP, EGP. Сравнительные характеристики.

Протокол RIP маршрутизации предназначен для сравнительно небольших и относительно однородных сетей (алгоритм Белмана-Форда).

Протокол OSPF (Open Shortest Pass First, RFC-1245-48, RFC-1583-1587, алгоритмы предложены Дикстрой) является альтернативой RIP в качестве внутреннего протокола маршрутизации. Переходные процессы в OSPF завершаются быстрее, чем в RIP.

Читай 4 вопроса далее. Смотри достоинства и недостатки.

24. Протокол SLIP.

SLIP - это сетевой протокол, позволяющий использовать для прямого выхода в Интернет обыкновенную телефонную линию и модем. Он требует специального программного обеспечения и работает совместно с протоколом TCP/IP как протокол более низкого уровня.

Протокол SLIP (Serial Line IP, RFC-1055) - это простейший способ инкапсуляции IP-дейтограмм для последовательных каналов связи. Этот протокол стал популярным благодаря возможностям подключения домашних персональных машин к сети Интернет через порт RS-232, который соединен с модемом.

Протокол SLIP выполняет единственную функцию - он позволяет в потоке бит, которые поступают по выделенному (или коммутируемому) каналу, распознать начало и конец IP-пакета. Помимо протокола IP, другие протоколы сетевого уровня

SLIP не поддерживает. IP-дейтограмма в случае SLIP должна завершаться специальным символом 0xC0 называемым конец. Во многих реализациях дейтограмма и начинается с этого символа. Если какой-то байт дейтограммы равен символу конец, то вместо него передается двухбайтовая последовательность 0xDB, 0xDC. Октет 0xDB выполняет в SLIP функцию ESC-символа. Если же байт дейтограммы равен 0xDB, то вместо него передается последовательность 0xDB, 0xDD. Использование протокола SLIP предполагает выполнение ряда условий:

- Каждый партнер обмена должен знать IP-адрес своего адресата, так как не существует метода обмена такого рода информацией.
- SLIP в отличии от Ethernet не использует контрольных сумм, поэтому обнаружение и коррекция ошибок целиком ложится на программное обеспечение верхних уровней.
- Так как кадр SLIP не имеет поля тип, его нельзя использовать, в отличии от кадров Ethernet, для реализации других протоколов методом инкапсуляции.

Ввиду его функциональной простоты, SLIP используется в основном на коммутируемых линиях связи, которые не характерны для ответственных и скоростных сетевых соединений. Тем не менее коммутируемый канал отличается от некоммутируемого только более низким качеством и необходимостью выполнять процедуру вызова абонента, поэтому SLIP вполне применим и на выделенных каналах.

Подключение SLIP кардинально отличается от подключения Ethernet тем, что в "сети" присутствует всего два компьютера: SLIP-клиент (ваш компьютер) и SLIP-сервер. По этой причине соединение SLIP часто называется соединением "точка-точка" ("point-to-point" connection). Обобщение этой идеи под названием "протокол PPP" (Point to Point Protocol) также реализовано в системе Linux.

Впервые протокол SLIP был применен в 1984 году в 4.2BSD. Скорость передачи информации при использовании протокола SLIP не превышает 19.2кб/с, что обычно достаточно для интерактивного обмена в рамках протоколов telnet или RLOGIN. Максимальный размер передаваемого блока (MTU) для SLIP лежит вблизи 256-512 байт, что обеспечивает разумный компромисс между значением задержки отклика (~256мс.) и эффективностью использования канала (~98% для CSLIP). При этом для передачи одного символа (нажатая клавиша) используется 20 байт заголовка в IP-дейтограмме и 20 байт TCP-заголовка. Если мы учтем издержки формирования SLIP-кадра, накладные расходы превосходят 40 байт. Частично этот недостаток устранен в новой версии CSLIP (Compressed SLIP, RFC-1144, предложенной Джекобсоном в 1990 году). В CSLIP заголовок сокращается до 3-5байт (против 40 в SLIP). Эта версия протокола способна поддерживать до 16 TCP-соединений на каждом из концов последовательного канала. Многие современные SLIP-драйверы поддерживают и CSLIP.

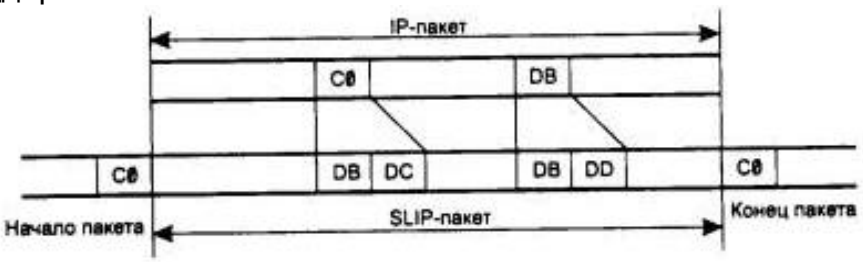


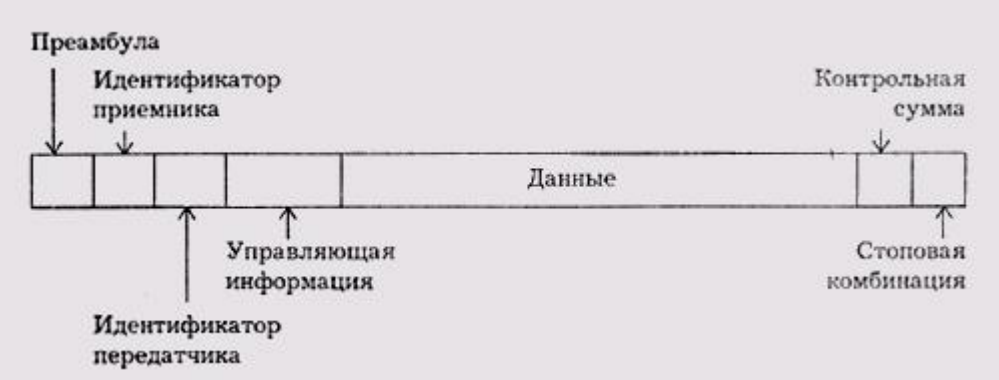
Рис. Инкапсуляция IP-пакетов в SLIP-пакеты

25. Понятие пакета, его структура. Технологии передачи пакетов в Ethernet.

Назначение пакетов и их структура. Информация в локальных сетях, как правило, передается отдельными порциями, кусками, называемыми в различных источниках пакетами, кадрами или блоками. Использование пакетов связано с тем, что в сети, как правило, одновременно может происходить несколько сеансов связи (во всяком случае, при топологиях «шина» и «кольцо»), то есть в течение одного и того же интервала времени могут идти два или больше процессов передачи данных между различными парами абонентов. Пакеты как раз и позволяют разделить во времени сеть между передающими информацию абонентами. Длина пакета зависит от типа сети, но обычно она составляет от нескольких десятков байт до нескольких килобайт. Важно также и то, что при передаче больших массивов информации становится довольно высокой вероятность ошибки из-за помех и сбоев. Обнаружить ошибку в массиве из нескольких мегабайт намного сложнее, чем в пакете из нескольких килобайт. С другой стороны, пакеты имеют преимущества и перед побайтовой (8 бит) или пословной (16 бит или 32 бита) передачей информации, так как увеличивается полезная загрузка сети за счет уменьшения требуемого количества служебной информации.

Существует некоторая оптимальная длина пакета (или оптимальный диапазон длин пакетов), при которой средняя скорость обмена информацией по сети будет максимальна. Структура пакета определяется прежде всего аппаратурными особенностями данной сети, выбранной топологией и типом среды передачи информации, а также существенно зависит от используемого протокола (порядка обмена информацией). Но существуют некоторые общие принципы формирования пакета, определяемые характерными особенностями обмена информацией по любым локальным сетям.

Типичная структура пакета



- Стартовая комбинация, или преамбула, которая обеспечивает настройку аппаратуры адаптера или другого сетевого устройства на прием и обработку пакета. Это поле может отсутствовать или сводиться к одному-единственному стартовому биту.
- Сетевой адрес (идентификатор) принимающего абонента, то есть индивидуальный или групповой номер, присвоенный каждому принимающему абоненту в сети. Этот адрес позволяет приемнику распознать пакет, адресованный ему лично, группе, в которую он входит, или всем абонентам сети одновременно.
- Сетевой адрес (идентификатор) передающего абонента, то есть индивидуальный или групповой номер, присвоенный каждому передающему абоненту. Этот адрес информирует принимающего абонента, откуда пришел данный пакет. Включение в пакет адреса передатчика необходимо в том случае, когда

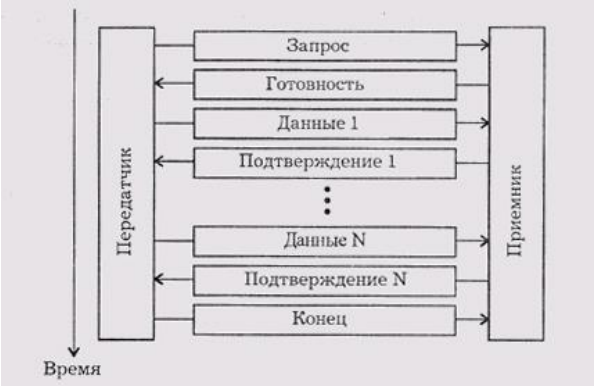
одному приемнику могут попеременно приходить пакеты от разных передатчиков.

- Служебная информация, которая указывает на тип пакета, его номер, размер, формат, маршрут его доставки, на то, что с ним надо делать приемнику и т.д.
- Данные - та информация, ради передачи которой используется данный пакет. Правда, существуют специальные управляющие пакеты, которые не имеют поля данных. Их можно рассматривать как сетевые команды. Пакеты, включающие поле данных, называются информационными пакетами. Управляющие пакеты могут выполнять функцию начала сеанса связи, конца сеанса связи, подтверждения приема информационного пакета, запроса информационного пакета и т.д.
- Контрольная сумма пакета - это числовой код, формируемый передатчиком по определенным правилам и содержащий в свернутом виде информацию обо всем пакете. Приемник, повторяя вычисления, сделанные передатчиком, с принятым пакетом, сравнивает их результат с контрольной суммой и делает вывод о правильности или ошибочности передачи пакета. Если пакет ошибочен, то приемник запрашивает его повторную передачу.
- Стоповая комбинация служит для информирования аппаратуры принимающего абонента об окончании пакета, обеспечивает выход аппаратуры приемника из состояния приема. Это поле может отсутствовать, если используется самосинхронизирующийся код, позволяющий детектировать факт передачи пакета.

Нередко в структуре пакета выделяют всего три поля:

- Начальное управляющее поле пакета (или заголовок пакета), то есть поле, включающее в себя стартовую комбинацию, сетевые адреса приемника и передатчика, а также служебную информацию.
- Поле данных пакета.
- Конечное управляющее поле пакета (или заключение, трейлер), включающее в себя контрольную сумму и стоповую комбинацию, а также, возможно, служебную информацию.

Помимо термина «пакет» в литературе также используется термин «кадр». Иногда под этими терминами имеется в виду одно и то же, но иногда подразумевается, что кадр вложен в пакет. Пример обмена пакетами при сеансе связи:



26. Маршрутизация: основные понятия, уровни маршрутизации.

См. 7 вопрос. А лучше лекции.

27. Протокол FTP. Модель, основные команды, безопасность, производительность.

FTP-архивы являются одними из основных информационных ресурсов Internet. Фактически, это распределенный депозитарий текстов, программ, фотографий и прочей информации, хранящейся в виде файлов на различных компьютерах во всем мире.

Технология FTP была разработана в рамках проекта ARPA и предназначена для обмена большими объемами информации между машинами с различной архитектурой. Главным в проекте было обеспечение надежной передачи, поэтому с современной точки зрения FTP кажется перегруженным излишними редко используемыми возможностями. Стержень технологии составляет FTP-протокол.

FTP-протокол.

FTP (File Transfer Protocol, или “Протокол передачи данных”) - один из старейших протоколов в Internet и входит в его стандарты. Первые спецификации FTP относятся к 1971 году. FTP предназначен для решения задач разделения доступа к файлам на удаленных хостах, прямого или косвенного использования ресурсов удаленных компьютеров, обеспечения независимости клиента от файловых систем удаленных хостов, эффективной и надежной передачи данных.

Обмен данными в FTP происходит по TCP-каналу. Обмен построен на технологии “клиент-сервер”. FTP не может использоваться для передачи конфиденциальных данных, поскольку не обеспечивает защиты передаваемой информации и передает между сервером и клиентом открытый текст. FTP-сервер может потребовать от FTP-клиента аутентификации (т.е. при присоединении к серверу FTP-пользователь должен будет ввести свой идентификатор и пароль). Однако пароль, и идентификатор пользователя будут переданы от клиента на сервер открытым текстом.

Модели работы FTP.

Простейшая модель работы протокола FTP представлена на рисунке 1.

В FTP соединение инициируется интерпретатором протокола пользователя. Команды FTP генерируются интерпретатором протокола пользователя и передаются на сервер. Ответы сервера отправляются пользователю также по каналу управления. В общем случае пользователь имеет возможность установить контакт с интерпретатором протокола сервера и отличными от интерпретатора протокола пользователя средствами.

Команды FTP определяют параметры канала передачи данных и самого процесса передачи. Они также определяют и характер работы с удаленной и локальной файловыми системами.

Сессия управления инициализирует канал передачи данных. При организации канала передачи данных последовательность действий другая, отличная от организации канала управления. В этом случае сервер инициирует обмен данными в соответствии с согласованными в сессии управления параметрами.

Канал данных устанавливается для того же хоста, что и канал управления, через который ведется настройка канала данных. Канал данных может быть использован как для приема, так и для передачи данных.

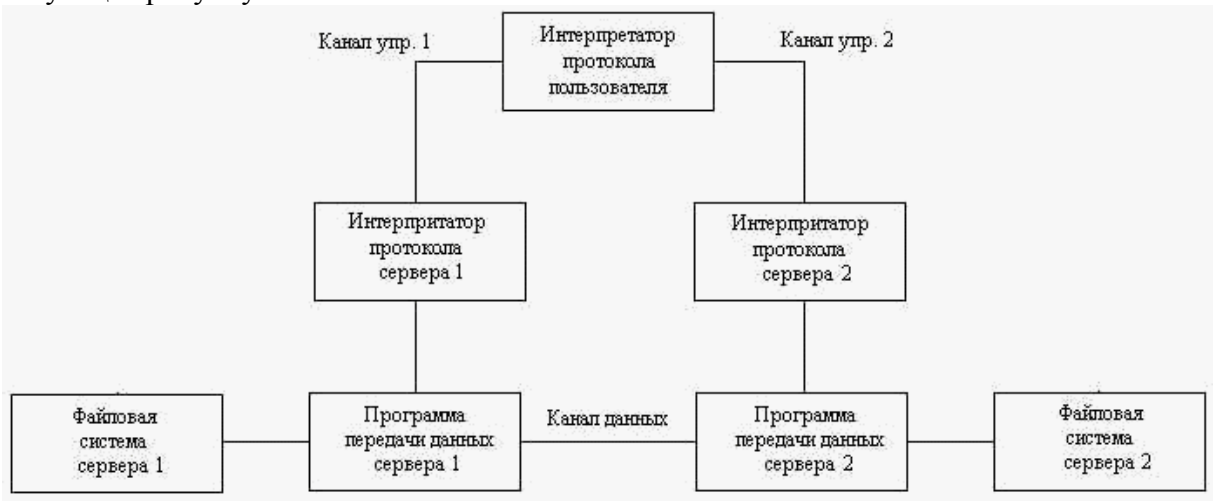
Алгоритм работы протокола FTP состоит в следующем:
Сервер FTP использует в качестве управляющего соединение на TCP порт 21, который всегда находится в состоянии ожидания соединения со стороны пользователя FTP.
После того как устанавливается управляющее соединение модуля “Интерпретатор протокола пользователя” с модулем сервера — “Интерпретатор протокола сервера”, пользователь (клиент) может отправлять на сервер команды. FTP-команды определяют параметры соединения передачи данных: роль участников соединения (активный или пассивный), порт соединения (как для модуля “Программа передачи данных пользователя”, так и для модуля “Программа передачи данных сервера”), тип передачи, тип передаваемых данных, структуру данных и управляющие директивы, обозначающие действия, которые пользователь хочет совершить (например, сохранить, считать, добавить или удалить данные или файл и другие).
После того как согласованы все параметры канала передачи данных, один из участников соединения, который является пассивным (например, “Программа передачи данных пользователя”), становится в режим ожидания открытия соединения на заданный для передачи данных порт. После этого активный модуль (например, “Программа передачи данных сервера”) открывает соединение и начинает передачу данных.
После окончания передачи данных, соединение между “Программой передачи данных сервера” и “Программой передачи данных пользователя” закрывается, но управляющее соединение “Интерпретатора протокола сервера” и “Интерпретатора протокола пользователя”

остается открытым. Пользователь, не закрывая сессии FTP, может еще раз открыть канал передачи данных.

Возможна ситуация, когда данные могут передаваться на третью машину. В этом случае пользователь организует канал управления с двумя серверами и прямой канал данных между ними. Команды управления идут через пользователя, а данные - напрямую между серверами. Канал управления должен быть открыт при передаче данных между машинами. Иначе, в случае его закрытия передача данных прекращается.

Алгоритм работы при соединение двух FTP-серверов, ни один из которых не расположен на локальном хосте пользователя:
Модуль “Интерпретатор протокола пользователя” указал модулю сервера “Интерпретатор протокола сервера 1” работать в пассивном режиме, после чего модуль “Интерпретатор протокола сервера 1” отправил пользователю адрес и номер порта (N) , который он будет слушать.
Модуль “Интерпретатор протокола пользователя” назначил модуль сервера 2 “Интерпретатор протокола сервера 2” в качестве активного участника соединения и указал ему передавать данные на хост “Интерпретатор протокола сервера 1” на порт (N).
“Интерпретатор протокола пользователя” подал “Интерпретатору протокола сервера 1” команду “сохранить поступившие данные в таком-то файле”, а “Интерпретатор протокола сервера 2” — “передать содержимое такого-то файла”.
Между модулями “Интерпретатор протокола сервера 1” и “Интерпретатор протокола сервера 2” образуется поток данных, который управляется клиентским хостом.

Ниже приведена схема организации передачи данных между двумя серверами FTP, соответствующая рисунку 2.



Здесь использованы следующие обозначения: User PI - интерпретатор протокола пользователя; Server1(2) интерпретатор протокола сервера1 (сервера2).

User PI (U) ЫI Server1 (S1)	User PI (U) ЫI Server2 (S2)
U Ю S1: Connect U Ю S1: PASV U Ь S1: 227 Entering Passive Mode. A1, A2, A3, A4, a1, a2	U Ю S2 Connect U Ю S2: PORT A1, A2, A3, A4, a1, a2

	U Ь S2: 200 Okay
U Ю S1: STOR ...	U Ю S2: RETR ...
S1 Ю S2: Connect ...	

Основу передачи данных FTP составляет механизм установления соединения между соответствующими портами и выбора параметров передачи. Каждый участник FTP-соединения должен поддерживать порт передачи данных по умолчанию. По умолчанию “Программа передачи данных пользователя” использует тот же порт, что и для передачи команд (обозначим его “U”), а “Программа передачи данных сервера” использует порт L-1, где “L”- управляющий порт. Однако, участниками соединения используются порты передачи данных, выбранные для них “Интерпретатором протокола пользователя”, поскольку из управляющих процессов участвующих в соединении, только “Интерпретатор протокола пользователя” может изменить порты передачи данных как у “Программы передачи данных пользователя”, так и у “Программы передачи данных сервера”.

Пассивная сторона соединения должна до того, как будет подана команда “начать передачу”, “слушать” свой порт передачи данных. Активная сторона, подающая команду к началу передачи данных, определяет направление перемещения данных.

После того как соединение установлено, между “Программой передачи данных сервера” и “Программой передачи данных пользователя” начинается передача. Одновременно по каналу “Интерпретатор протокола сервера” — “Интерпретатор протокола пользователя” передаются уведомления о получении данных. Протокол FTP требует, чтобы управляющее соединение было открыто, пока по каналу обмена данными идет передача. Сессия FTP считается закрытой только после закрытия управляющего соединения.

Как правило, сервер FTP ответственен за открытие и закрытие канала передачи данных.
Сервер FTP должен самостоятельно закрыть канал передачи данных в следующих случаях:
Сервер закончил передачу данных в формате, который требует закрытия соединения.
Сервер получил от пользователя команду “прервать соединение”.
Пользователь изменил параметры порта передачи данных.
Было закрыто управляющее соединение.
Возникли ошибки, при которых невозможно возобновить передачу данных.

Команды протокола.

Команды управления контролем передачи данных, которыми обмениваются “Интерпретатор протокола сервера” и “Интерпретатор протокола пользователя”, можно разделить на три большие группы:
Команды управления доступом к системе.
Команды управления потоком данных.
Команды FTP-сервиса.

Рассмотрим несколько наиболее характерных команд из каждой группы. Среди команд управления доступом к системе следует отметить следующие:

USER. Как правило, эта команда открывает сессию FTP между клиентом и сервером. Аргументом команды является имя (идентификатор) пользователя для работы с файловой системой. Эта команда может подаваться не только в начале, но и в середине сессии, если, например, пользователь желает изменить идентификатор, от имени которого будут проводиться действия. При этом все переменные, относящиеся к старому идентификатору, осво-

бождаются. Если во время изменения идентификатора происходит обмен данными, обмен завершается со старым идентификатором пользователя.

PASS. Данная команда подается после ввода идентификатора пользователя и, в качестве аргумента содержит пароль пользователя. Напомним, что данные аутентификации FTP передаются по сети открытым текстом, поэтому для обеспечения защищенности канала пользователю необходимо предпринимать дополнительные меры.

CWD. Команда позволяет пользователям работать с различными каталогами удаленной файловой системы. Аргументом команды является строка, указывающая путь каталога удаленной файловой системы, в котором желает работать пользователь.

REIN. Команда реинициализации. Эта команда очищает все переменные текущего пользователя, сбрасывает параметры соединения. Если в момент подачи команды происходит передача данных, передача продолжается и завершается с прежними параметрами.

QUIT. Команда закрывает управляющий канал. Если в момент подачи команды происходит передача данных, канал закрывается после окончания передачи данных.

Команды управления потоком устанавливают параметры передачи данных. Все параметры, описываемые этими командами имеют значение по умолчанию, поэтому команды управления потоком используются только тогда, когда необходимо изменить значение параметров передачи, используемых по умолчанию. Команды управления потоком могут подаваться в любом порядке, но все они должны предшествовать командам FTP-сервиса. Из команд управления потоком данных следует выделить следующие:

PORT. Команда назначает адрес и порт хоста, который будет использоваться как активный участник соединения по каналу передачи данных. Аргументами команды являются 32-битный IP адрес и 16-битный номер порта соединения. Эти значения разбиты на шесть 8-битных полей и представлены в десятичном виде: h1, h2, h3, h4, p1, p2, где hN - байты адреса (от старшего к младшему), а pN - байты порта (от старшего к младшему).

PASV. Эта команда отправляется модулю, который будет играть пассивную роль в передаче данных (“слушать” соединение). Ответом на данную команду должна быть строка, содержащая адрес и порт хоста, находящиеся в режиме ожидания соединения в формате команды PORT — “h1, h2, h3, h4, p1, p2”.

Команды TYPE, STRU, MODE определяют, соответственно, тип передаваемых данных (ASCII, Image и другие), структуру или формат передачи данных (File, Record, Page), способ передачи (Stream, Block и другие). Использование этих команд очень важно при построении взаимодействия в гетерогенных средах и весьма отличающихся операционных и файловых систем взаимодействующих хостов.

Команды FTP-сервиса определяют действия, которые необходимо произвести с указанными файлами. Как правило, аргументом команд этой группы является путь к файлу. Синтаксис указанного пути должен удовлетворять требованиям формата файловой системы обработчика команды. Из команд FTP-сервиса можно выделить следующие:

RETR. Эта команда указывает модулю “Программа передачи данных сервера” передать копию файла, заданного параметром этой команды, модулю передачи данных на другом конце соединения.

STOR. Команда указывает модулю “Программа передачи данных сервера” принять данные по каналу передачи данных и сохранить их как файл, имя которого задано параметром этой команды. Если такой файл уже существует, он будет замещен новым, если нет, будет создан новый.

Команды RNFR и RNTO должны следовать одна за другой. Первая команда содержит в качестве аргумента старое имя файла, вторая - новое. Последовательное применение этих команд переименовывает файл.

ABOR. Команда предписывает серверу прервать выполнение предшествующей сервисной команды (например, передачу файла) и закрыть канал передачи данных.

Команда DELE удаляет указанный файл.

Команды MKD и RMD, соответственно, создают и удаляют указанный в аргументе каталог.

При помощи команд LIST и NLST можно получить список файлов в указанном каталоге.

Все команды FTP-протокола отправляются “Интерпретатором протокола пользователя” в текстовом виде - по одной команде в строке. Каждая строка команды - идентификатор и аргументы - заканчиваются символами <CRLF>. Имя команды отделяется от аргумента символом пробела - <SP>.

Обработчик команд возвращает код обработки каждой команды, состоящий из трех цифр. Коды обработки составляют определенную иерархическую структуру и, как правило, определенная команда может возвратить только определенный набор кодов. За кодом обработки команды следует символ пробела - <SP>, затем следует текст пояснения. Например, строка успешного завершения операции выглядит следующим образом: “200 Command okay”.

Ниже приведен пример работы с FTP-протокола. Обозначения: S - сервер, U - пользователь.

S: 220 Service ready for new user
U: USER Gluk
> S: 331 User name okay, need password
U: PASS murmur
S: 230 User logged in, proceed
U: RETR test.txt
S: 150 File status okay; about to open data connection

<Идет передача файла ...>

S: 226 Closing data connection, file transfer successful
U: TYPE I
S: 200 Command okay
U: STOR /home/images/first.my
S: 550 Access denied
U: QUIT

28. NFS, RPC и XDR.

NFS

NFS (network file system, sun microsystems, RFC-1094) обеспечивает прозрачный доступ к удаленным файлам так, что с точки зрения программиста эти файлы выглядят, как местные. При этом даже в написании имен файлов никак не проявляется их истинное местонахождение. NFS является частью операционной системы. Различие работы с местными и удаленными файлами проявляется лишь на системном уровне. Пользователь может почувствовать различие лишь по времени выполнения соответствующих операций обмена. nfs поддерживает операции по созданию, переименованию, копированию и стиранию файлов или каталогов и т.д.

Основой системы NFS является вызов удаленных процедур RPC, схема взаимодействия "клиент-сервер". NFS-сервер получает запросы от клиента в виде UDP-дейтограмм через порт 2049 (Рис. 4.5.16.1).

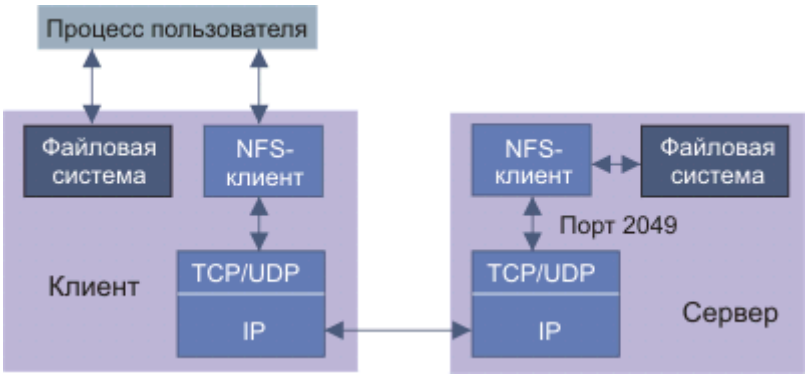


Рис. 4.5.16.1. Схема реализации NFS-системы клиент-сервер

RPC

RPC (Remote Procedure Call, RFC-1057) процедура, разработанная SUN microsystem, в настоящее время используется практически во всех системах, базирующихся на UNIX. RPC - это программа, которая реализует вызов удаленных подпрограмм, способствуя построению распределенных программ. Она позволяет программе, называемой клиентом, послать сообщение серверу. Далее программа-клиент ожидает сообщения-отклика. RPC работает совместно с универсальной системой представления внешней информации XDP (External Data Representation). Сообщение запрос содержит параметры, которые определяют, что должно быть сделано на удаленной ЭВМ. В свою очередь отклик несет в себе информацию о результатах выполнения запроса.

RPC может работать как на TCP, так и UDP транспортных уровнях. Использование RPC-техники упрощает программирование, так как не требует написания сетевых программ. Если используется протокол UDP, все что связано с обработкой тайм-аутов, повторных пересылок и пр. спрятано в внутри системных RPC-модулей. Формат RPC-запроса для UDP-версии показан на рис. 4.5.16.2.

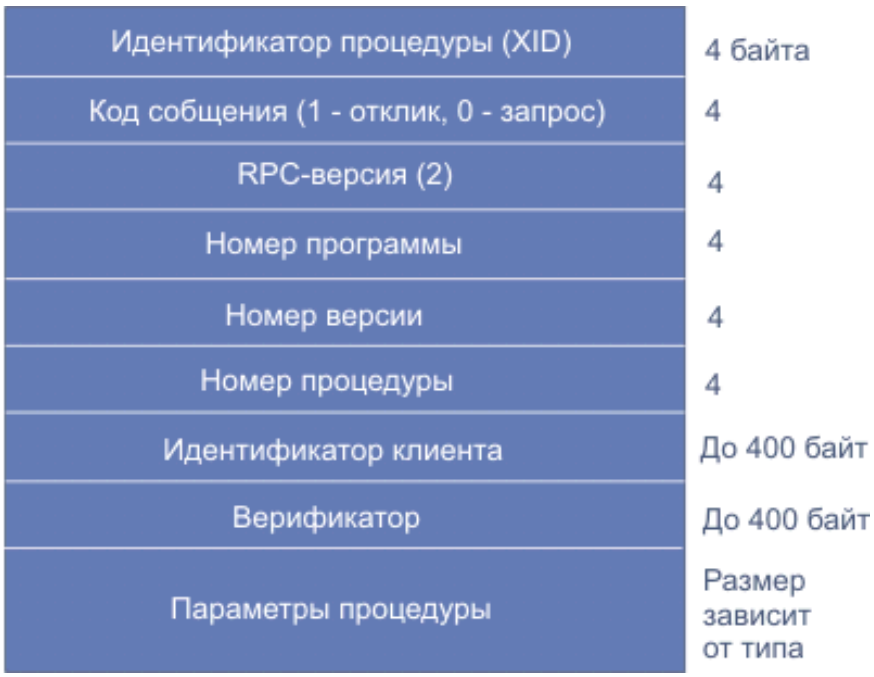


Рис. 4.5.16.2. Формат RPC-запроса

Поле *идентификатор процедуры* устанавливается программой-клиента, пакет-отклик использует тот же идентификатор, что позволяет контролировать их соответствие. Каждый новый RPC-запрос имеет новый идентификатор. В настоящее время номер версии rpc равен 2. Следующие три поля содержат переменные: номер программы, номер версии и номер процедуры, которые определяют тип запрашиваемой клиентом процедуры. В поле *идентификатор клиента* может быть записан цифровой код клиента, идентификатор группы или вообще ничего. Поле *верификатор* используется при пересылке зашифрованных сообщений. Формат параметров процедуры зависит от типа этой процедуры. Размер поля параметров равен длине UDP-дейтограммы минус сумма длин остальных полей, включая верификатор. В случае работы с TCP-сегментами, где длина пакета не определена, между TCP-заголовком и XID вводится 4-х октетное поле длины RPC-сообщения. Формат RPC-отклика для UDP-версии (Рис. 4.5.16.3):

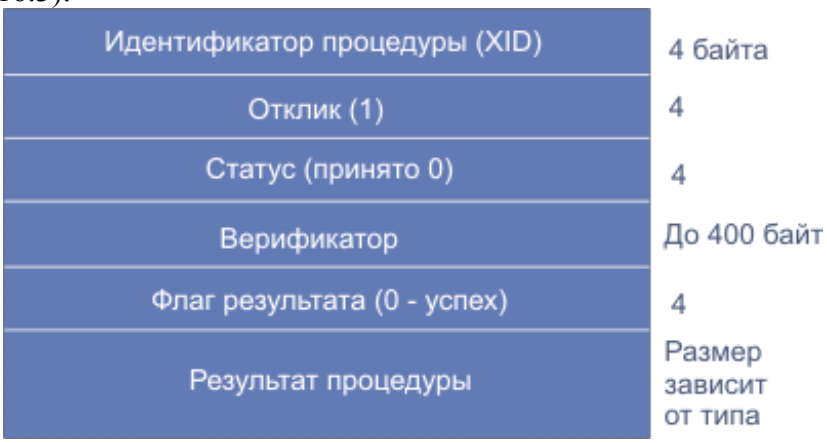


Рис. 4.5.16.3. Формат RPC-отклика

Поле *отклик*, содержащее 1, указывает на то, что данное сообщение представляет собой отклик на поступивший ранее запрос. Поле *статус* содержит 0 в случае, если запрос воспринят. Запрос игнорируется при конфликте кодов RPC-версии или неудачной идентификации клиента. Поле *флаг* результата принимает значение 0 при успешной обработке запроса. Ненулевое значение этого поля указывает на ошибку.

Для записи параметров RPC-запросов, откликов, параметров и результатов выполнения процедуры используется внешнее представление данных (XDR - External Data

Representation, RFC-1014). Отправитель, формируя RPC-сообщение, использует XDR-формат, а получатель преобразует данные из этого формата в традиционное представление.

Существует два базовых вида отклика: MSG_ACCEPTED (сообщение принято) и MSG_DENIED (не принято). Факт приема сообщения не означает выполнение запрошенных процедур, поэтому клиенту выдается дополнительная информация о результатах взаимодействия его запроса с удаленной системой. RPC может найти применение при построении больших распределенных информационных систем, баз данных и систем управления. XDR позволяет программисту избежать написания специальных программ преобразования. Например, в разных ЭВМ используются различные форматы представления чисел с плавающей запятой. XDP берет на себя согласование форматов и делает написание прикладных программ машинно-независимым.

Программы RPC-сервера используют большое число портов с нестандартизованными номерами.

Раздел 3

1. Сетевые системы хранения данных: Протокол Serial ATA.

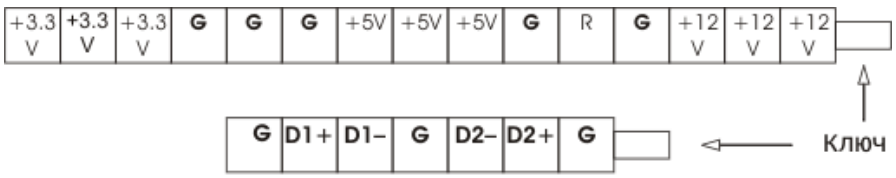
SATA (англ. Serial ATA) — последовательный интерфейс обмена данными с накопителями информации (как правило, с жёсткими дисками). SATA является развитием интерфейса ATA (IDE), который после появления SATA был переименован в PATA.
ATA - Advanced Technology Attachment

Преимущества (перед PATA):

- (главное) использование последовательной шины вместо параллельной. (меньше наводок, больше износостойкость, и т.п.)
- Стандарт SATA предусматривает горячую замену устройств и функцию очереди команд
- На шине располагается 1 устройство (=> выше скорость).

Разъемы

SATA использует 7-контактный разъём вместо 40-контактного разъёма у PATA. Разъём питания SATA подаёт 3 напряжения питания: +12 В, +5 В и +3,3 В; однако современные устройства могут работать без напряжения +3,3 В, что даёт возможность использовать пассивный переходник с стандартного разъёма питания IDE на



G – заземление; R – зарезервировано;
D1+,D1–,D2+,D2– два канала передачи данных (от контроллера к устройству и от устройства к контроллеру соответственно). Для передачи сигнала используется технология LVDS, провода каждой пары (D1+,D1– и D2+, D2–) являются экранированными витыми парами.

SATA/150

Стандарт SATA предусматривал работу шины на частоте 1,5 ГГц, обеспечивающей пропускную способность приблизительно в 1,2 Гбит/с (150 Мб/с). (20%-я потеря производительности объясняется использованием системы кодирования 8B/10B). Пропускная способность SATA/150 незначительно выше пропускной способности шины Ultra ATA (UDMA/133).

SATA/300 (SATA II)
Стандарт SATA/300, работающий на частоте 3 ГГц и обеспечивающий пропускную способность до 2,4 Гбит/с (300 Мб/с) Теоретически SATA/150 и SATA/300 устройства должны быть совместимы (контроллер <=> устройство), однако для некоторых устройств и контроллеров требуется ручное выставление режима работы (джампер).

2. Сетевые системы хранения данных: Протокол Parallel ATA.

ATA - Advanced Technology Attachment. (он же IDE, UDMA(??) и ATAPI)
IDE - Integrated Drive Electronics

Используется 40/80 - проводный шлейф (во втором случае каждый 2ой проводник заземлен). Макс. длина = 46 см.

На шине может быть до 2х устройств (+одно устройство в режиме read-only)

Стандарт	Проп. спос. стандарт(МБ/сек)	Макс. размер диска	Свойства
ATA 1 (ATA, IDE)	PIO (3.3, 5.2, 8.3) DMA (2.1, 4.2, 8.3)	137 Гб	28-bit LBA, CHS*
ATA 2 (Fast ATA, Fast IDE)	PIO (11.1, 16.6) DMA (13.3, 16,6)	то же	
ATA 3 (EIDE)	то же	то же	S.M.A.R.T., Security
ATA/ATAPI-4	Ultra DMA/33 (16.7, 25.0, 33.3)		Support for CD-ROM, etc., via ATAPI packet commands
ATA/ATAPI-5	Ultra DMA 66 (44.4, 66.7)		80-wire cables
ATA/ATAPI-6	UDMA 5 (100)	128 ПиБ (дофига)	48-bit LBA Automatic Acoustic Management
ATA/ATAPI-7	UDMA 6 (133)		
ATA/ATAPI-8	under construction		

* Спецификация ATA предусматривала 28-битный режим адресации. Это позволяло адресовать 228 (268 435 456) секторов по 512 байт каждый, что давало максимальную ёмкость в 137 ГБ (128 ГиБ). В стандартных PC BIOS поддерживал до 7,88 ГиБ (8,46 ГБ), допуская максимум 1024 цилиндра, 256 головок и 63 сектора. Это ограничение на число цилиндров/головок/секторов CHS(cylinder/head/sector) в сочетании со стандартом IDE привело к ограничению адресуемого пространства в 504 МиБ (528 МБ). Для преодоления этого ограничения была введена схема адресации LBA(logical block address), что позволило адресовать до 7,88 ГиБ. Со временем и это ограничение было снято, что позволило адресовать сначала 32 ГиБ, а затем и все 137 ГБ, предусмотренные на то время спецификацией ATA.

PIO – Program in-out режим обмена, при котором данные переписывает ЦПУ
DMA – режим обмена информацией, при котором данные переписывает отдельный контроллер (контроллер DMA)

формат пакета

3. Сетевые системы хранения данных: Протокол iSCSI.

iSCSI (Internet Small Computer System Interface) — это протокол, который базируется на TCP/IP и разработан для установления взаимодействия и управления системами хранения данных, серверами и клиентами. При этом используется IP-адрес, TCP порт, SCSI узел. Обычно данному протоколу противопоставляют iFCP
iSCSI описывает:
1) Транспортный протокол для SCSI, который работает поверх TCP
2) Новый механизм инкапсуляции SCSI команд в IP сети

3) Протокол для новой генерации систем хранения данных, которые будут использовать «родной» TCP/IP

Противоречия и трудности

1) В IP пакеты доставляются получателю без соблюдения строгой последовательности, и ПК восстанавливает данные, на что затрачиваются определенные ресурсы (до 100% загрузки ПК с сетевой картой без аппаратной реализации TCP/IP при использовании iSCSI). В то же время, по спецификации SCSI, как канального интерфейса, все пакеты должны передаваться один за другим без задержки, а нарушение этого порядка приводит к потере данных.
2) Проблема большого времени задержки в ip сетях (75 микросекунд) => подходят только быстрые сети (напр. Gigabit Ethernet)

Преимущества

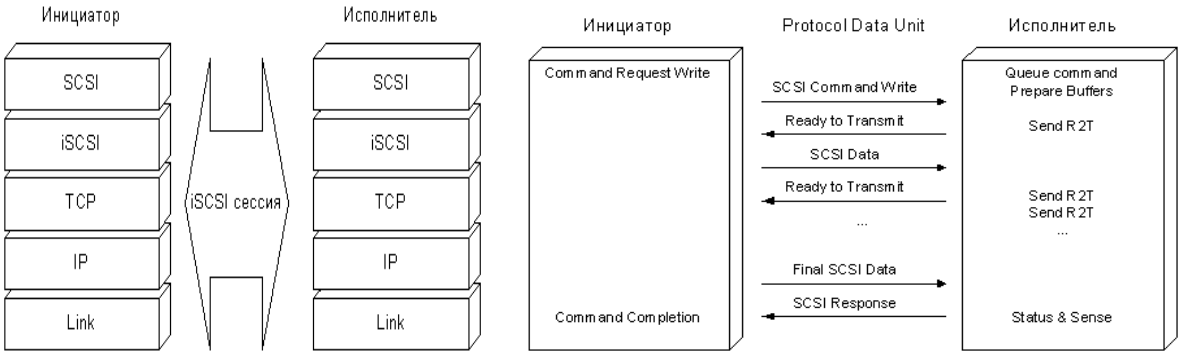
1) Географическое распределение данных (SAN)
2) Безопасность

iSCSI имеет четыре составляющие:

- Управление именами и адресами (iSCSI Address and Naming Conventions) (!имена устройств могут обрабатываться обычным DNS-сервером)
- Управление сеансом (iSCSI Session Management). (login/logout)
- Обработка ошибок (iSCSI Error Handling).
- Безопасность (iSCSI Security).

Обработка ошибок

Для того, чтобы обработка ошибок и восстановление после сбоев функционировали корректно, как инициатор, так и исполнитель должны иметь возможность буферизации команд до момента их подтверждения.



а) чтото типа OSI б) прохождение транзакции

4. Сетевые системы хранения данных: Протокол Parallel SCSI.

SCSI - Small Computer Systems Interface - интерфейс, разработанный для объединения на одной шине различных по своему назначению устройств. После стандартизации в 1986 году, SCSI начал широко применяться в компьютерах Apple Macintosh, Sun Microsystems. В компьютерах совместимых с IBM PC SCSI не пользуется такой популярностью в связи со своей сложностью и сравнительно высокой стоимостью.

Есть 3 основных стандарта SCSI, каждый из которых имеет множество дополнительных и необязательных возможностей. Некоторые комбинации возможностей имеют собственные наименования.

Interface	Bus	Clock	Max.	Max. cable	Max. number of

	width	speed	throughput	length	devices
SCSI	8-bit	5 MHz	5 MB/s	6 m	8
Fast SCSI	8-bit	10 MHz	10 MB/s	1.5-3 m	8
Fast-Wide SCSI	16-bit	10 MHz	20 MB/s	1.5-3 m	16
Ultra SCSI	8-bit	20 MHz	20 MB/s	1.5-3 m	5-8
Ultra Wide SCSI	16-bit	20 MHz	40 MB/s	1.5-3 m	5-8
Ultra2 SCSI	8-bit	40 MHz	40 MB/s	12 m	8
Ultra2 Wide SCSI	16-bit	40 MHz	80 MB/s	12 m	16
Ultra3 SCSI	16-bit	40 MHz DDR	160 MB/s	12 m	16
Ultra-320 SCSI	16-bit	80 MHz DDR	320 MB/s	12 m	16

Шина требует терминатор (бывает активный и пассивный);
Ultra-3 SCSI – одновременное использование фронтов и спадов импульсов, добавлена контрольная сумма (CRC) и исправление ошибок
Ultra-640 SCSI – малая длина кабеля, нераспространен

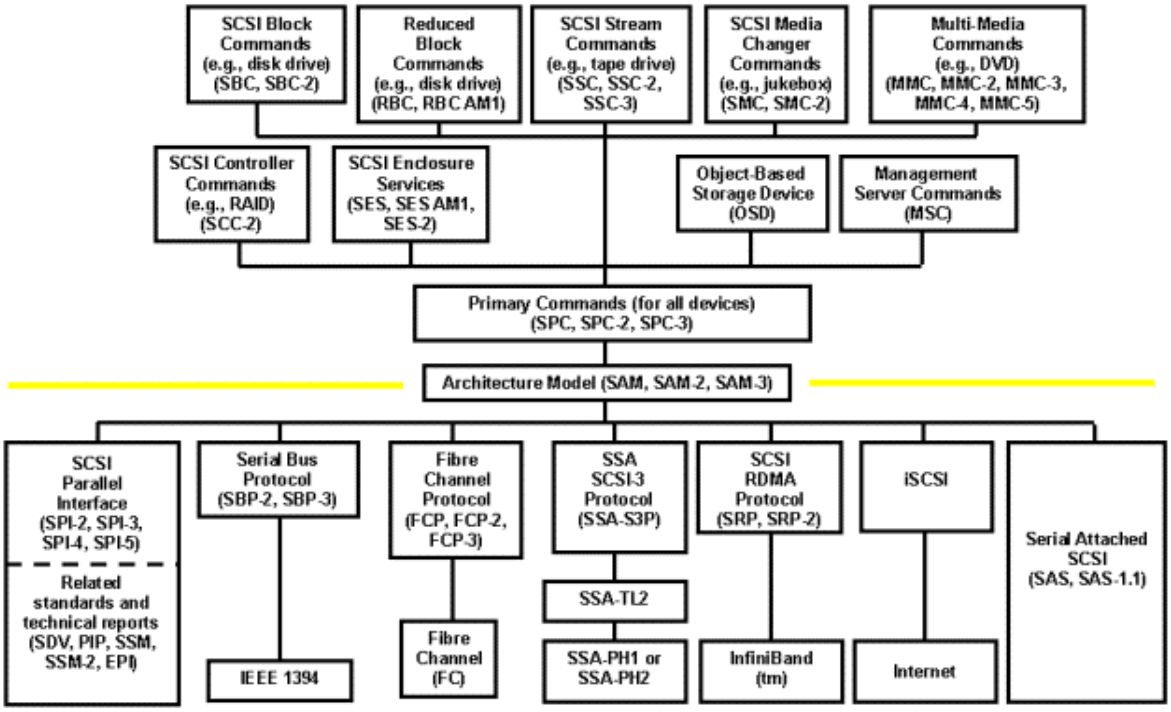
Протокол

Взаимодействие идёт между инициатором(от кого запрос) и целевым устройством(от кого ответ). На шине может быть не более 2х Контроллеров шины - Host Base Adapter (HBA);

Команды SCSI посылаются в виде блоков описания команды (Command Descriptor Block, CDB). Длина блока: 6, 10, 12, 16 байт. (в последних версиях – переменная длина) Блок состоит из кода команды(1байт) и параметров команды.

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code = 03h							
1	LUN			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Control							

Команды SCSI по смыслу (поле Control) делятся на четыре категории: N (non-data), W (запись данных от инициатора целевым устройством), R (чтение данных) и B (двусторонний обмен данными). Всего существует порядка 60 различных команд SCSI.



Верхняя часть рисунка – перечисление различных наборов команд.
Нижняя часть рисунка – перечисление модификаций протокола SCSI (не нужно здесь).

Устройство на SCSI-шине имеет как минимум один номер логического устройства - Logical Unit Number (LUN).

Устройства имеют приоритет (поле SCSI ID ??? лек.). Его размер – 8 или 16 бит (в зависимости от разрядности шины) Приоритеты в порядке уменьшения: 7,6,...0, 15,14,9,8.

После обработки команды цель возвращает Sense code – код завершения операции

1 00h OK
2 02h Check condition
3 04h Condition met
4 08h Busy
5 10h Intermediate
6 14h Intermediate - Condition met
7 18h Reservation conflict
8 22h Command terminated
9 28h Queue (task set) full
10 30h ACA active
11 40h Task aborted

5. Архитектура систем хранения данных NAS

Системы NAS

Устройства **NAS (Network Attached Storage)** – устройства хранения, подключённые напрямую в сеть. В отличие от других систем NAS обеспечивает файловый доступ к данным и никак иначе. NAS-устройства представляют из себя комбинацию системы хранения данных и сервера, к которому она подключена. В простейшем варианте обычный сетевой сервер, предоставляющий файловые ресурсы, является устройством NAS (**Рисунок 4**)

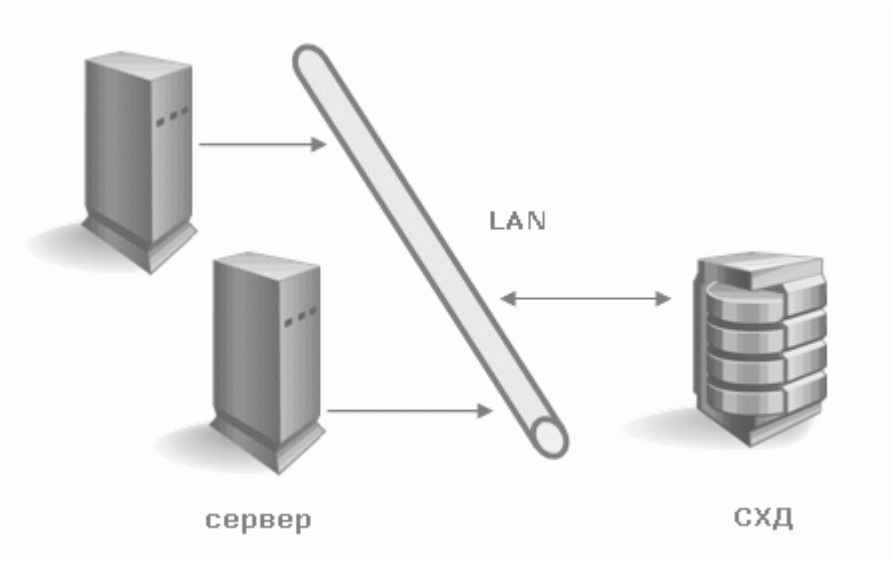


Рисунок 14 Архитектура NAS

Все минусы такой схемы аналогичны DAS-топологии, за некоторым исключением. Из добавившихся минусов отметим возросшую, и часто значительно, стоимость – правда, стоимость пропорциональна функциональности, а тут уже часто «есть за что платить». NAS-устройства могут быть простейшими «коробочками» с одним портом ethernet и двумя жёсткими дисками в RAID1, позволяющими доступ к файлам по лишь одному протоколу CIFS (Common Internet File System) до огромных систем в которых могут быть установлены сотни жёстких дисков, а файловый доступ обеспечивается десятком специализированных серверов внутри NAS-системы. Число внешних Ethernet-портов может достигать многих десятков, а ёмкость хранимых данных – несколько сотен терабайт (например EMC Celerra CNS). Такие модели по надёжности и производительности могут далеко обходить многие midrange-устройства SAN. Что интересно, NAS-устройства могут быть частью SAN-сети и не иметь собственных накопителей, а лишь предоставлять файловый доступ к данным, находящимся на блочных устройствах хранения. В таком случае NAS берёт на себя функцию мощного специализированного сервера, а SAN – устройства хранения данных, то есть мы получаем топологию DAS, скомпонованную из NAS- и SAN-компонентов.

NAS-устройства очень хороши в гетерогенной среде, где необходим быстрый файловый доступ к данным для многих клиентов одновременно. Также обеспечивается отличная надёжность хранения и гибкость управления системой вкупе с простотой обслуживания. На надёжности особо останавливаться не будем – этот аспект СХД рассмотрен выше. Что касается гетерогенной среды, доступ к файлам в рамках единой NAS-системы может быть получен по протоколам TCP/IP, CIFS, NFS, FTP, TFTP и другим, включая возможность работы NAS, как iSCSI-target, что обеспечивает функционирование с различным ОС, установленными на хостах. Что касается лёгкости обслуживания и гибкости управления, то эти возможности обеспечиваются специализированной ОС, которую трудно вывести из строя и не нужно обслуживать, а также простотой разграничения прав доступа к файлам. К примеру, возможна работа в среде Windows Active Directory с поддержкой требуемой функциональности – это может быть LDAP, Kerberos Authentication, Dynamic DNS, ACLs, назначение квот (quotas), Group Policy Objects и SID-истории. Так как доступ обеспечивается к файлам, а их имена могут содержать символы различных языков, многие NAS обеспечивают поддержку кодировок UTF-8, Unicode. К выбору NAS стоит подходить даже тщательнее, чем к DAS-устройствам, ведь такое оборудование может не поддерживать необходимые вам сервисы, например, Encrypting File Systems (EFS) от Microsoft и IPSec. К слову можно заметить, что NAS распространены намного меньше, чем устройства SAN, но процент таких систем всё же постоянно, хотя и медленно, растёт – в основном за счёт вытеснения DAS.

6. Сетевые системы хранения данных: Дисковые массивы: JBOD, RAID.

JBOD – Just a Bundle Of Discs
RAID – Redundant Array of Inexpensive (or Independent) Disks

RAID и JBOD могут быть как программными (драйверы в ОС), так и аппаратными (RAID контроллер)

JBOD

JBOD-массив не имеет избыточности. При выходе из строя одного из дисков в JBOD массиве, из строя выходит весь массив, и все данные, хранящиеся на нем, теряются. Типичное применение JBOD - просто объединение двух или большего количества дисков с маленькой емкостью в один большой (задумывался для рационального использования старых дисков – лек.). Отличие от RAID0 – нет выигрыша в скорости.
Разумен, если диски – старые и имеют разную емкость.

RAID

	N	Емк.	ЭХ	ОУ	RR	RW	SR	SW	$
0	2,3,4,…	S*N	100%	none	4	4	4,5	4	$
1	2	S*N/2	50%	4	3	3	2	3	$$
2	many	varies, large	~ 70-80%	2	2	1	4	2,5	$$$$$
3	3,4,5,...	S*(N-1)	(N-1)/N	3	3	1	4	2,5	$$
4	3,4,5,...	S*(N-1)	(N-1)/N	3	4	1,5	3	2	$$
5	3,4,5,...	S*(N-1)	(N-1)/N	3	4,5	2	3,5	2,5	$$
6	4,5,6,...	S*(N-2)	(N-2)/N	4,5	4,5	1	4	2	$$$
7	varies	varies	varies	3	4,5	4	4,5	4	$$$$$
01, 10	4,6,8,...	S*N/2	50%	4	4,5	3,5	4,5	3,5	$$$
03, 30	6,8,9,10,…	S*N0*(N3-1)	(N3-1)/N3	3,5	4	2	4,5	3	$$$$
05, 50	6,8,9,10,…	S*N0*(N5-1)	(N5-1)/N5	3,5	4,5	3	4	3	$$$$
15, 51	6,8,10,…	S*((N/2)-1)	((N/2)-1)/N	5	4	3	4	4	$$$$$

ЭХ – Эффективность Хранения; N – число дисков в массиве; N0 – число дисков, в измерении RAID0 (для RAID30); N3 – число дисков, в измерении RAID3(для RAID30); S – емкость самого маленького диска в массиве; ОУ – отказоустойчивость; RR/RW – производительность случайного чтения/записи; SR/SW – производительность послед. чтения/записи;

RAID 0 – "не избыточный" массив, выходящий из строя в случае выхода из строя любого диска. +) высокая скорость доступа -) низкая защита
RAID 1 – Теневой" диск представляет собой точную копию "основного".
RAID 2 – RAID0 + блокировка диска + синхр-я вращения (для старых HDD) (не используется – лек.)
RAID 3 – не используется (лек.)
RAID 4 – данные о четности хранятся на отдельном диске. не используется (лек.)
RAID 5 – данные о четности распределены по всем дискам => один диск всегда восстановим
RAID 6 – RAID 5 + честность пишется дважды
RAID XY = блоки RAID X, объединенные в RAID Y

7. Архитектура систем хранения данных DAS

Устройства **DAS (Direct Attached Storage)** – системы хранения, подключаемые напрямую к серверу. Сюда относятся как самые простые SCSI-системы, подключаемые к SCSI/RAID-контроллеру сервера, так и устройства FibreChannel, подключенные прямо к серверу, хотя и предназначены они для сетей SAN. В этом случае топология DAS является вырожденной SAN (сетью хранения данных) (**Рисунок 3**)

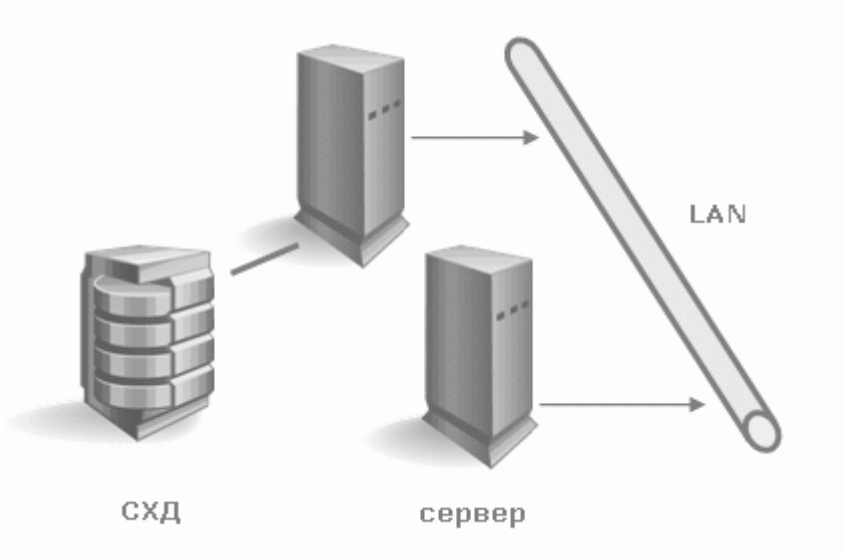


Рисунок 15 Архитектура DAS

В этой схеме один из серверов имеет доступ к данным, хранящимся на СХД. Клиенты получают доступ к данным, обращаясь к этому серверу через сеть. То есть сервер имеет блочный доступ к данным на СХД, а уже клиенты пользуются файловым доступом – эта концепция очень важна для понимания. Минусы такой топологии очевидны:

6. Низкая надежность – при проблемах сети или аварии сервера данные становятся недоступны всем сразу.
7. Высокая латентность, обусловленная обработкой всех запросов одним сервером и использующимся транспортом (чаще всего – IP).
8. Высокая загрузка сети, часто определяющая пределы масштабируемости путём добавления клиентов.
9. Плохая управляемость – вся ёмкость доступна одному серверу, что снижает гибкость распределения данных.
10. Низкая утилизация ресурсов – трудно предсказать требуемые объёмы данных, у одних устройств DAS в организации может быть избыток ёмкости (дисков), у других её может не хватать – перераспределение часто невозможно или трудоёмко.

8. Сетевые системы хранения данных: Оптические и магнито-оптические устройства хранения данных.

теория

Магнитооптика (МО) - раздел физики, в котором изучаются изменения оптических свойств сред под действием магнитного поля и обусловленные этим изменения взаимодействия оптического излучения с помещенным в поле веществом.

Характеристики	**MO**	**CD-RW**	**DVD**	**Дискета**	**Стримерная лента**	**JAZ**	**ZIP**
Проблема хранения		Солн. свет		Размагничивание, различные влияния	Застревание и разрыв		Влияние полей
Срок хранения: - Гарантия - Теория	50 150	50 100	50 100	5 15	20 40	10 50	8 46
Проблемы с драйверами	+	-	+	-	-	-	-
Ошибки записи	-	+	-	+	+	-	+
Циклы перезаписи	10000000	1000	1000	100-200	800	10000	1000
Максимальная емкость (Гб)	9,1 (5,25)* 2,6 (3,5)	700[iv]	8 (16)	0	много	??	??
Цена устройства (в среднем, $)	400	200	400	20	800	500	150

* 5.25, 3.5 – размер привода в дюймах

Технология	**Использование**	**Емкость**	**Скорость доступа**	**Ожидаемые затраты (накопитель/носитель)**
CD-RW	Настольные системы	650 Мбайт	120 мс	400 долларов/от 12 до 15 долларов
DVD+RW	Настольные системы Сетевая среда (после появления библиотек)	8 Гбайт	100 мс	От 700 долларов до 800 долларов/от 25 до 40 долларов
DVD-RAM	Настольные системы Сетевая среда (после появления библиотек)	8 Гбайт	200 мс	От 700 до 800 долларов/ от 25 до 40 долларов
MO	Сетевая среда	5,2 Гбайт	35 мс	От 1500 до 3000 долларов/75 долларов

МО-диск представляет собой поликарбонатную подложку (частот его также называют слоем) толщиной 1,2 мм, на которую нанесено несколько тонкопленочных слоев, в котором заключается магнитная часть технологии, а оптическая представлена считывающим лазером.

Интерфейсы подключения МО приводов – ATAPI, SCSI, LPT, USB 1.1/2.0, IEEE 1394, и т.д.
Сменный МО диск – примерно 1.8 дискеты по размеру

Осн. параметры (лек.): скорость вращения (RPM), время доступа, интерфейс, стоимость хранения 1Гб

9. Архитектуры систем хранения данных: Сравнительные характеристики DAS, NAS, SAN, рекомендации по применению.

См. вопрос 1.4

10. Сетевые системы хранения данных: Дисковые массивы с RAID: уровни RAID, принципы организации по уровням.

RAID 0 представлен как дисковый массив повышенной производительности и меньшей отказоустойчивости.
RAID 1 определён как зеркальный дисковый массив.
RAID 2 зарезервирован для массивов, которые применяют код Хемминга.
RAID 3 и 4 используют массив дисков с чередованием и выделенным диском чётности.
RAID 5 используют массив дисков с чередованием и "невыделенным диском чётности".
RAID 6 используют массив дисков с чередованием и двумя независимыми "чётностями" блоков.
RAID 10 — RAID 0, построенный из RAID 1 массивов
RAID 50 — RAID 0, построенный из RAID 5
RAID 60 - RAID 0, построенный из RAID 6

11. Сети хранения данных – основные понятия, определения и термины. Дисковые устройства хранения данных.

Сети хранения данных довольно объемны и позволяют хранить сотни гигабайт или терабайт информации.
В целях улучшения качества доступа к системе, а также для более удобной замены тех или иных модулей системы, сети хранения данных отделены от серверов (компьютеров) и представляют собой самостоятельные устройства, снабженные большим количеством жестких дисков.
Каждый из таких дисков, в случае появления дефекта, может быть извлечен и заменен прямо в ходе работы, без необходимости прерывать процесс. С целью дальнейшего увеличения надежности хранения данных системы обеспечивают сохранение информации одновременно на многих дисках.
Таким образом, компания, имеющая верно подобранную и грамотно структурированную сеть хранения данных, защищена от потери информации в результате таких происшествий как перебои в системе электропитания, пожары или наводнения.

Сеть хранения данных (SAN)

Сеть хранения данных - это сервера, рабочие станции, дисковые хранилища и ленточные библиотеки, связанные по протоколу Fibre Channel (FC). Fibre Channel - это протокол, оптимизированный для быстрой передачи сообщений (100 MB/s), позволяющий передавать информацию на расстояние в десятки километров (с использованием оптической среды передачи) и поддерживающий такие протоколы верхних уровней, как SCSI, TCP/IP, ATM, HIPPI, ESCON и VI. Использование SAN по сравнению к существующему подходу хранения данных, где данные хранятся во внутренних хранилищах, присоединенных к отдельным серверам, дает ряд значительных

преимуществ: Доступность - одни и те же данные могут быть доступны в один момент нескольким серверам и рабочим станциям. Надежность - надежный механизм транспортировки (протокол FC), возможность дублирования путей передачи и хранения (RAID-массивы) данных уменьшает вероятность возникновения ошибок и дает возможность их исправления. Масштабируемость - серверы и хранилища данных могут быть добавлены в сеть независимо друг от друга. Управляемость - адимнистрирование хранилищ данных может производится централизованно из «единой точки». Производительность - данные и служебная информация передаются раздельно: передача данных происходит внутри SAN, передача служебной информации - внутри LAN, что значительно уменьшает вероятность возникновения перегрузки сети. Стоимость владения системой - механизмы обеспечения надежности и средства прогнозирования возникновения ошибок значительно уменьшают возможное время простоя информационной системы. Возможность централизованного управления уменьшает требования к количеству персонала , обслуживающего систему. Независимая инсталляция в SAN хранилищ данных и серверов позволяет начально построить систему с минимальными затратами и наращивать ее по мере надобности.

Основные составляющие SAN

- Внешние хранилища данных - дисковые массивы
    - Все дисковые массивы поддерживают возможность дублирования и горячей замены источников питания, RAID-контроллеров, интерфейсных модулей и жестких дисков, имеют развитые средства управления и мониторинга, а также возможность интегрирования в кластерные решения, как неотъемлемый компонент.
    - Ленточные Библиотеки (Tape Libraries) Применяются для архивирования информации, причем ресурсы локальной сети не используются - передача данных при резервном копировании происходит внутри SAN. Такой подход получил название LAN-Free-Backup.
- Адаптеры Fibre Channel

Подключение к SAN серверов и рабочих станций осуществляется посредством Fibre Channel-адаптеров. В настоящее время компания QLogic предлагает ряд однопортовых и двухпортовых FC-адаптеров для подключения как к медной, так к и оптической среде передачи. При использовании медной среды передачи максимальное расстояние между сервером и FC-концентратором равно 30 м. При использовании оптики оно может быть увеличено до 10 км.

- Коммуникационное оборудование SAN

Также как в случае локальных сетей для организации SAN (за исключением простейших случаев) необходимы FC-концентраторы (FC-Hubs), FC-коммутаторы (Switched Fabrics), а также FC-шлюзы (FC-bridges).

- Кабели и медиа-конвертеры

Для организации физического подключения оборудования в SAN применяются медные и оптические кабели, а также медиа-конвертеры, когда в сети используется разнородная среда передачи данных.

- Программное обеспечение

Для организации SAN и достижения максимальной выгоды от использования этой технологии применяется специализированное программное обеспечение. Такое ПО по исполняемым задачам можно класифицировать следующим образом: Управление и мониторинг состоянием устройств SAN Обеспечение совместного доступа нескольких компьютеров к общему дисковому массиву Разграничение доступа компьютеров к дисковым массивам Резервное копирование и архивирование данных

Дисковые устройства.

Для того, чтобы лучше понять особенности автономных накопителей, остановимся немного на одной из более простых технологий построения систем хранения данных — шинно-ориентированной технологии. Она предусматривает использование корпуса для дисковых накопителей и контроллера PCI RAID.

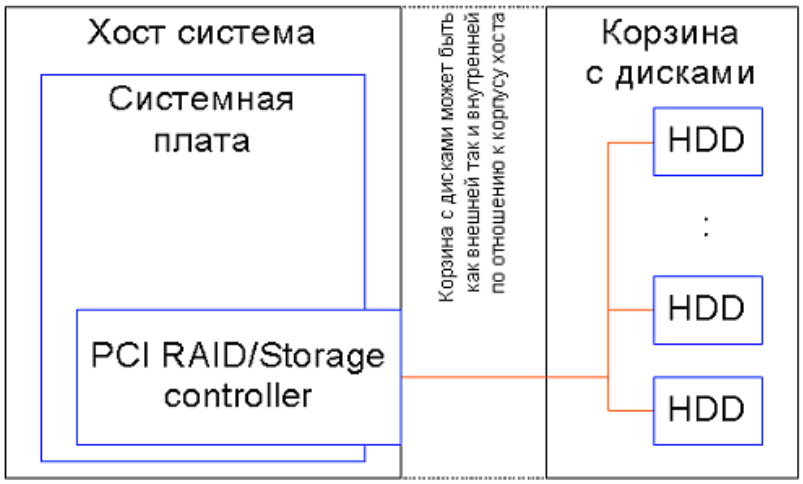


Рисунок 1. Шинно-ориентированная технология постоения систем хранения данных

Таким образом, между дисками и PCI-шиной хоста (от, англ. Host — в данном случае автономный компьютер, например сервер или рабочая станция) есть только один контроллер, который в значительной мере и задает быстродействие системы. Накопители, построенные по этому принципу, являются наиболее производительными. Но в связи с архитектурными особенностями практическое их использование, за исключением редких случаев, ограничивается конфигурациями с одним хостом.

К недостаткам шинно-ориентированной архитектуры накопителей следует отнести:
эффективное использование только в конфигурациях с одним хостом;
зависимость от операционной системы и платформы;
ограниченную масштабируемость;
ограниченные возможности по организации отказоустойчивых систем.

Естественно, всё это неважно, если данные нужны для одного сервера или рабочей станции. Наоборот, в такой конфигурации вы получите максимальное быстродействие за минимальные деньги. Но если вам нужна система хранения данных для большого вычислительного центра или даже для двух серверов, которым нужны одни и те же данные, шинно-ориентированная архитектура совершенно не подходит. Недостатков этой архитектуры позволяет избежать архитектура автономных дисковых подсистем. Основной принцип ее построения достаточно прост. Контроллер, который управляет системой, переносится из хост-компьютера в корпус накопителя, обеспечивая независимое от хост-систем функционирование. Следует отметить, что такая система может иметь большое количество внешних каналов ввода/вывода, что обеспечивает возможность подключения к системе нескольких, или даже многих компьютеров.

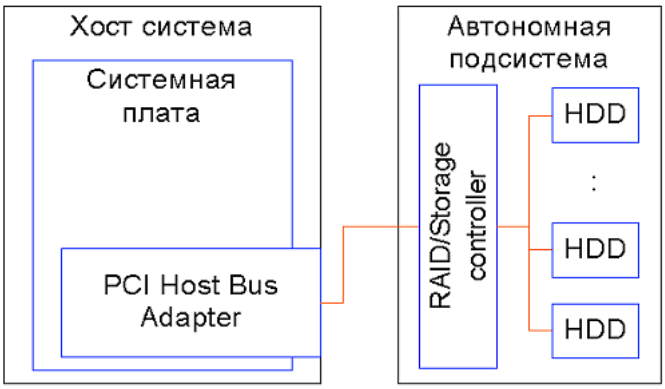


Рисунок 2. Автономная система хранения данных

Любая интеллектуальная система хранения данных состоит из аппаратной части и программного кода. В автономной системе всегда есть память, в которой хранится программа алгоритмов работы самой системы и процессорные элементы, которые этот код обрабатывают. Такая система функционирует независимо от того, с какими хост-системами она связана. Благодаря своей интеллектуальности автономные накопители зачастую самостоятельно реализуют множество функций по обеспечению сохранности и управлению данными. Одна из самых важных базовых и практически повсеместно используемых функций — это RAID (Redundant Array of Independent Disks). Другая, принадлежащая уже системам среднего и высокого уровня - это виртуализация. Она обеспечивает такие возможности как мгновенная копия или удаленное резервирование, а также другие, достаточно изощрённые алгоритмы.

12. Сети хранения данных – основные понятия, определения и термины. Ленточные устройства хранения данных.

Стри́мер[1] (от англ. streamer), также ле́нточный накопи́тель — запоминающее устройство на принципе магнитной записи на ленточном носителе, с последовательным доступом к данным, по принципу действия аналогичен бытовому магнитофону.

Основное назначение: запись и воспроизведение информации, архивация и резервное копирование данных.

Параметры: скорость доступа, скорость чтения/записи, емкость, алгоритмы сжатия, кол-во циклов перезаписи и т.д. см. ниже

+) низкая цена; большие объемы хранения информации; срок хранения информации выше, чем у HDD; аппаратное сжатие

-) никакая скорость доступа; мало циклов перезаписи;
типы
Digital Data Storage (DDS),
Digital Linear Tape (DLT),
Linear Tape-Open (LTO),
Advanced Intelligent Tape (AIT)

DLT
Разработан Hewlett-Packard. Использует пленку, изготовленную по технологии DAT (Digital Audio Tape) Ширина пленки 3.8 мм. Количество перезаписей: 100

Format	Date	Tape Length (m)	Capacity (GB)	Speed (MB/s)
DDS-1	1989	60/90	1.3/2.0	0.6
DDS-2	1993	120	4.0	0.6
DDS-3	1996	125	12.0	1.1
DDS-4	1999	150	20.0	2.4
DAT 72	2003	170	36.0	3.5
DAT 160	2006	N/A	N/A	N/A

LTO

Обязательно использует сжатие

Есть две модификации: 1) Ultrium (The high capacity) 2) Accelis (The high speed)

Характеристики: 1 million passes; 30 years of archival storage; 20000 loads and unloads; Maximum rewind time 98 seconds;

Generation	Date	Uncompressed capacity (GB)	Speed (MBytes/s)	**WORM** availability
LTO1	1999	100	20	No
LTO2	2002	200	40	No
LTO3	2005	400	80	Yes
LTO4	TBA (2006)	800	120	Yes
LTO5	TBA (2008)	1600	180	Yes
LTO6	TBA (2010)	3200	270	Yes

WORM - Write Once Read Many

13. Протокол FCP (Fibre Channel Protocol)

ibre Channel имеет уникальную систему физического интерфейса и форматы кадров, которые позволяют этому стандарту обеспечить простую стыковку с канальными протоколами IPI (Intelligent Peripheral Interface), SCSI, HIPPI, ATM, IP и 802.2. Это позволяет рганизовать скоростной канал между ЭВМ и дисковой накопительной системой RAID.

Быстродействие сетей Fibre Channel	n x 100Мбайт/с
Длины канала	10 км и более.
Предельная скорость передачи	4,25 Гбод.

В качестве транспортной среды может использоваться одномодовое или мультимодовое оптическое волокно. Допускается применение медного коаксиального кабеля и скрученных пар (при скоростях до 200 Мбайт/с).

Fibre Channel имеет шесть независимых классов услуг (каждый класс представляет определенную стратегию обмена информацией), которые облегчают решать широкий диапазон прикладных задач:

Класс 1 — Соединение с коммутацией каналов по схеме точка-точка (end-to-end) между портами типа n_port Класс удобен для аудио и видео приложений После установления соединения используется вся доступная полоса пропускания канала. При этом гарантируется, что кадры будут получены в том же порядке, в каком они были посланы.

Класс 2 — Обмен без установления соединения с коммутацией пакетов, гарантирующий доставку данных. соединение не устанавливается, порт может взаимодействовать одновременно с любым числом портов типа n_port, получая и передавая кадры. Нет гарантии того, что кадры будут доставлены в порядке передачи (за

исключением случаев соединения точка-точка или арбитражное кольцо). Используются схемы управления потоком буфер-буфер и точка-точка. (характерен для сетей, где время доставки данных не критично).

Класс 3 — Обмен дейтограммами без установления соединения и без гарантии доставки. Схема управления потоком буфер-буфер. Применяется для каналов scsi.

Класс 4 — Обеспечивает выделение определенной доли пропускной способности канала с заданным значением качества обслуживания (QoS). Работает только с топологией структура (fabric), где соединяются два порта типа n_port. Формируется два виртуальных соединения, обслуживающих встречные потоки данных. Пропускная способность этих соединения может быть различной. Как и в классе 1, здесь гарантируется порядок доставки кадров. Допускается одновременное соединение более чем с одним портом типа n_port. Используется схема управления потоком буфер-буфер. Каждое виртуальное соединение управляется независимо с помощью сигнала-примитива fc_rdy.

Класс 5 — Предполагает изохронное обслуживание. Регламентирующие документы находятся в процессе подготовки.

Класс 6 — Предусматривает мультикастинг-обслуживание в рамках топологии типа структура (fabric). При этом используется стандартный адрес 0xfffff5. n_port становится членом мультикаст-группы путем регистрации по адресу 0xfffff8.

Fibre Channel использует пакеты переменной длины (до 2148 байт), содержащие до 2112 байт данных. Такая длина пакета заметно снижает издержки, связанные с пересылкой заголовков (эффективность 98%) Только FDDI превосходит Fibre Channel по этому параметру (99%). В отличие от других локальных сетей, использующих 6-октетные адреса, fibre channel работает с 3-байтовыми адресами, распределяемыми динамически в процессе выполнения операции login.

Адрес 0xffffff зарезервирован для широковещательной адресации.

Адреса в диапазоне 0xfffff0-0xfffffe выделены для обращения к "структуре" (fabric), мультикастинг-серверу и серверу псевдонимов (alias-server).

n_port передает кадры от своего source_id (s_id) к destination_id (d_id). До выполнения операции fabric login s_id порта не определено. В случае арбитражного кольца применяются 3-октетные адреса al_pa, задаваемые при инициализации кольца. Для однозначной идентификации узлов используются 64-битовые имена-идентификаторы.

Формат пакетов в сетях Fibre Channel

Используются 24-битовые адреса, что позволяет адресовать до 16 миллионов объектов. Схемы сети:

- точка-точка,
- кольцевая архитектура с возможностью арбитража (FC-al) К кольцу может быть подключено до 128 узлов. Это - самое дешевое подключение
- «ткань соединений» (fabric), допускающее большое число независимых обменов одновременно).

Пятиуровневая модель

Протокол Fibre Channel предусматривает 5 уровней, которые определяют физическую среду, скорости передачи, схему кодирования, форматы пакетов, управление потоком и различные виды услуг.

На физическом уровне (FC-ph 1993 год) предусмотрены три подуровня. FC использует оптические волокна диаметром 62,5, 50мкм и одномодовые. Для обеспечения безопасности предусмотрен опционный контроль подключенности оптического разъема (OFC). Для этого передатчик время от времени посылает короткие световые импульсы приемнику. Если приемник получает такой импульс, процесс обмена продолжается.

FC-0 определяет физические характеристики интерфейса и среды, включая кабели, разъемы, драйверы (ECL, LED, лазеры), передатчики и приемники. Вместе с FC-1 этот уровень образует физический слой.

FC-1 определяет метод кодирования/декодирования (8B/10B) и протокол передачи, где объединяется пересылка данных и синхронизирующей информации.

FC-2 определяет правила сигнального протокола, классы услуг, топологию, методику сегментации, задает формат кадра и описывает передачу информационных кадров.

FC-3 определяет работу нескольких портов на одном узле и обеспечивает общие виды сервиса.

FC-4 обеспечивает реализацию набора прикладных команд и протоколов вышележащего уровня (например, для SCSI, IPI, IEEE 802, SBCCS, HIPPI, IP, ATM и т.д.)

Кольцевая архитектура

Прежде чем использовать кольцо его нужно инициализировать (процедура LIP), так чтобы каждый порт получил свой физический адрес al_pa (al_pa - один октет, что и определяет максимальное число портов в кольце арбитража). Процедура инициализации начинается сразу после включения питания посылкой сигнала-примитива LIP через порт l_port. Затем осуществляется выбор устройства, которое будет управлять процессом выбора al_pa.

Не используется маркерная схема доступа. Когда подключенное к сети устройство готово передать данные, он передает сигнал-примитив ARBX, где X - физический адрес устройства в кольце арбитража (al_pa). Если устройство получит свой собственный сигнал-примитив ARBX, оно получает контроль над кольцом и может начать передачу. Инициатор обмена посылает сигнал-примитив open (OPN) и устанавливает связь с адресатом. Время удержания контроля над кольцом не лимитируется. Если контроль над кольцом одновременно пытаются захватить два устройства, сравниваются значения X сигналов ARB. Устройство с меньшим al_pa получает преимущество, прибор с большим al_pa блокируется.

Перед передачей октеты преобразуются в 10-битовые кодовые последовательности, называемые символами передачи (кодировка IBM 8B/10B). Логической единице соответствует больший уровень световой энергии. В Fibre Channel предусмотрено два режима обмена буфер-буфер и точка-точка (end-to-end). Передача данных осуществляется только когда принимающая сторона готова к этому. Прежде чем что-либо посылать стороны должны выполнить операцию login. В ходе выполнения login определяется верхний предел объема пересылаемых данных (credit). Значение параметра credit задает число кадров, которые могут быть приняты. После передачи очередного кадра значение credit уменьшается на единицу. Когда значение этой переменной достигает нуля, дальнейшая передача блокируется до тех пор, пока получатель не обработает один или более кадров и будет готов продолжить прием.

Пример задачи:

4. Разработать bat (batch) файл. Содержащий команды управления, которые позволяют вывести на экран имя и адрес локального сервера разрешения сетевых имен (привести листинг с типовыми ответами сервера).
5. Разработать bat (batch) файл, содержащий команды управления, которые позволяют получить параметры конфигурации хоста, включая IP-адрес, маску подсети и шлюз по умолчанию, отобразить полную информацию о настройке параметров, освободить IP-адрес для указанного адаптера, обновить IP-адрес для указанного адаптера, очистить кэш разрешений (DNS), обновить все DHCP-аренды и перерегистрировать DNS-имена, отобразить содержимое кэша разрешений (DNS), отобразить все допустимые для этого адаптера коды (IDs) DHCP-классов, изменить код (ID) DHCP-класса (привести листинг с типовыми ответами сервера).

2. Понятие MAC адреса, его структура.

MAC-адрес (от англ. Media Access Control — управление доступом к носителю) — это уникальный идентификатор, сопоставляемый с различными типами оборудования для компьютерных сетей. Большинство сетевых протоколов канального уровня используют одно из трёх пространств MAC-адресов, управляемых IEEE: MAC-48, EUI-48 и EUI-64. Адреса в каждом из пространств теоретически должны быть глобально уникальными. Не все протоколы используют MAC-адреса, и не все протоколы, использующие MAC-адреса, нуждаются в подобной уникальности этих адресов.

В широковещательных сетях (таких, как сети на основе Ethernet) MAC-адрес позволяет уникально идентифицировать каждый узел сети и доставлять данные только этому узлу. Таким образом, MAC-адреса формируют основу сетей на канальном уровне, которую используют протоколы более высокого (сетевого) уровня. Для преобразования MAC-адресов в адреса сетевого уровня и обратно применяются специальные протоколы (например, ARP и RARP в сетях TCP/IP).

Адреса типа MAC-48 наиболее распространены; они используются в таких технологиях, как Ethernet, Token ring, FDDI и др. Они состоят из 48 бит, таким образом, адресное пространство MAC-48 насчитывает 248 (или 281 474 976 710 656) адресов. Согласно подсчётам IEEE, этого запаса адресов хватит по меньшей мере до 2100 года.

EUI-48 от MAC-48 отличается лишь семантически: в то время как MAC-48 используется для сетевого оборудования, EUI-48 применяется для других типов аппаратного и программного обеспечения.

Идентификаторы EUI-64 состоят из 64 бит и используются в FireWire, а также в IPv6 в качестве младших 64 бит сетевого адреса узла.

Структура MAC-адреса

Стандарты IEEE определяют 48-разрядный (6 октетов) MAC-адрес, который разделен на четыре части.

Первые 3 октета (в порядке их передачи по сети; старшие 3 октета, если рассматривать их в традиционной бит-реверсной шестнадцатиричной записи MAC-адресов) содержат 24-битный уникальный идентификатор организации (OUI)[1], который производитель полу-

чает в IEEE. При этом используются только младшие 22 разряда (бита), 2 старшие имеют специальное назначение:
первый бит указывает, для одиночного (0) или группового (1) адресата предназначен кадр
следующий бит указывает, является ли MAC-адрес глобально (0) или локально (1) администрируемым.

Следующие три октета выбираются изготовителем для каждого экземпляра устройства.

Таким образом, глобально администрируемый MAC-адрес устройства глобально уникален и обычно «зашит» в аппаратуру.

Администратор сети имеет возможность, вместо использования «зашитого», назначить устройству MAC-адрес по своему усмотрению. Такой локально администрируемый MAC-адрес выбирается произвольно и может не содержать информации об OUI. Признаком локально администрируемого адреса является соответствующий бит первого октета адреса (см. выше).

Для того, чтобы узнать MAC-адрес сетевого устройства используются следующие команды:
Windows - ipconfig /all
Windows - getmac

3. Понятие пакета, его структура. Технологии передачи пакетов в Ethernet.

IP (англ. Internet Protocol — межсетевой протокол) — маршрутизируемый сетевой протокол, основа стека протоколов TCP/IP.

Протокол IP (RFC 791) используется для негарантированной доставки данных (разделяемых на так называемые *пакеты*) от одного узла сети к другому. Это означает, что на уровне этого протокола (третий уровень сетевой модели OSI) не даётся гарантий надёжной доставки пакета до адресата. В частности, пакеты могут прийти не в том порядке, в котором были отправлены, оказаться повреждёнными или не прибыть вовсе. Гарантии безошибочной доставки пакетов дают протоколы более высокого (транспортного) уровня сетевой модели OSI — например, TCP — которые используют IP в качестве транспорта.

В современной сети Интернет используется IP четвёртой версии, также известный как IPv4. В протоколе IP этой версии каждому узлу сети ставится в соответствие IP-адрес длиной 4 октета (иногда говорят «байта», подразумевая распространённый восьмибитовый минимальный адресуемый фрагмент памяти ЭВМ). При этом компьютеры в подсетях объединяются общими начальными битами адреса. Количество этих бит, общее для данной подсети, называется маской подсети (ранее использовалось деление пространства адресов по классам — A, B, C; класс сети определялся диапазоном значений старшего октета и определял число адресуемых узлов в данной сети, сейчас используется бесклассовая адресация).

В настоящее время вводится в эксплуатацию шестая версия протокола — IPv6, которая позволяет адресовать значительно большее количество узлов, чем IPv4. Эта версия отличается повышенной разрядностью адреса, встроенной возможностью шифрования и некоторыми другими особенностями. Переход с IPv4 на IPv6 связан с трудоёмкой работой операторов связи и прозводителей программного обеспечения и не может быть выполнен одномоментно. На начало 2007 года в Интернете присутствовало около 760 сетей, работающих по протоколу IPv6. Для сравнения, на то же время в адресном пространстве IPv4 присутствовало более 203 тысяч сетей, но в IPv6 сети гораздо более крупные, нежели в IPv4.

IP-пакет

IP-пакет — форматированный блок информации, передаваемый по вычислительной сети. Соединения вычислительных сетей, которые не поддерживают пакеты, такие как традиционные соединения типа «точка-точка» в телекоммуникациях, просто передают данные в виде последовательности байтов, символов или битов. При использовании пакетного форматирования сеть может передавать длинные сообщения более надежно и эффективно.

Структура IP-датаграммы (пакета)

В протоколе четвертой версии (IPv4)

0								1								2								3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Версия				**IHL**				**Тип обслуживания**								**Длина пакета**															
Идентификатор																**Флаги**				**Смещение фрагмента**											
Число переходов (TTL)								**Протокол**								**Контрольная сумма заголовка**															
IP-адрес отправителя (32 бита)																															
IP-адрес получателя (32 бита)																															

Параметры (до 320 бит)	Данные (до 65535 байт минус заголовок)

- Версия — для IPv4 значение поля должно быть равно 4.
- IHL — длина заголовка IP-пакета в 32-битных словах (dword). Именно это поле указывает на начало блока данных в пакете. Минимальное корректное значение для этого поля равно 5.
- Идентификатор — значение, назначаемое отправителем пакета и предназначенное для определения корректной последовательности фрагментов при сборке датаграммы.
- 3 бита флагов. Первый бит должен быть всегда равен нулю, второй бит DF (don't fragment) определяет возможность фрагментации пакета и третий бит MF (more fragments) показывает, не является ли этот пакет последним в цепочке пакетов.
- Смещение фрагмента — значение, определяющее позицию фрагмента в потоке данных.
- Протокол — идентификатор интернет-протокола следующего уровня (см. IANA protocol numbers и RFC 1700). В IPv6 называется «Next Header».

В протоколе 6 версии (IPv6)

Версия (4 бита)	Класс трафика (8 бит)	Метка потока (20 бит)	
Длина полезной нагрузки (16 бит)		След. заголовок (8 бит)	Число переходов
IP-адрес отправителя (128 бит)			
IP-адрес получателя (128 бит)			
Данные			

- Версия — для IPv6 значение поля должно быть равно 6.
- Класс трафика — определяет приоритет трафика (QoS, класс обслуживания).
- Метка потока — уникальное число, одинаковое для однородного потока пакетов.
- Длина полезной нагрузки — длина данных (заголовок IP-пакета не учитывается).
- Следующий заголовок — Определяет следующий инкапсулированный протокол.
- Число переходов — максимальное число роутеров, которые может пройти пакет. При прохождении роутера это значение уменьшается на единицу и по достижению нуля пакет отбрасывается.

4. Понятие фреймов Ethernet (IEEE 802.3 Packet Framing), изменения в Ethernet II.

5. Сетевые службы и сервисы. Понятие и основные характеристики.

6. Разрешение сетевых имен с помощью DNS. Протокол ARP.

DNS (англ. Domain Name System — система доменных имён) — распределённая система преобразования имени хоста (компьютера или другого сетевого устройства) в IP адрес. DNS работает в сетях TCP/IP. Как частный случай, DNS может хранить и обрабатывать и обратные запросы, определения имени хоста по его IP адресу.

Ключевые характеристики DNS

DNS обладает следующими характеристиками:

- Распределённость хранения информации. Каждый узел сети в обязательном порядке должен хранить только те данные, которые входят в его зону ответственности и (возможно) адреса корневых DNS-серверов.
- Кеширование информации. Узел может хранить некоторое количество данных не из своей зоны ответственности для уменьшения нагрузки на сеть.
- Иерархическая структура, в которой все узлы объединены в дерево, и каждый узел может или самостоятельно определять работу нижестоящих узлов, или делегировать (передавать) их другим узлам.

- Резервирование. За хранение и обслуживание своих узлов (зон) отвечают (обычно) несколько серверов, разделённые как физически, так и логически, что обеспечивает сохранность данных и продолжение работы даже в случае сбоя одного из узлов.

DNS важна для работы Интернета, ибо для соединения с узлом необходима информация о его IP-адресе, а для людей проще запоминать буквенные (обычно осмысленные) адреса, чем последовательность цифр IP-адреса. В некоторых случаях это позволяет использовать виртуальные серверы, например, HTTP-сервера, различая их по имени запроса. Первоначально преобразование между доменными и IP-адресами производилось с использованием специального текстового файла HOSTS, который составлялся централизованно и обновлялся на каждой из машин сети вручную. С ростом Сети возникла необходимость в эффективном, автоматизированном механизме, которым и стала DNS.

DNS была разработана Полом Мокапетрисом в 1983 году; оригинальное описание механизмов работы описано в RFC 882 и RFC 883. В 1987 публикация RFC 1034 и RFC 1035 изменили спецификацию DNS и отменили RFC 882 и RFC 883 как устаревшие. Некоторые новые RFC дополнили и расширили возможности базовых протоколов.

ARP (англ. *Address Resolution Protocol* — протокол разрешения адресов) — сетевой протокол, предназначенный для преобразования IP-адресов (адресов сетевого уровня) в MAC-адреса (адреса канального уровня) в сетях TCP/IP. Он определён в RFC 826.

ARP (протокол разрешения адресов) - очень распространённый и чрезвычайно важный протокол. Каждый узел сети имеет два адреса, физический адрес и логический адрес. В сети Ethernet для идентификации источника и получателя информации используются оба адреса. Информация пересылаемая от одного компьютера другому по сети содержит в себе физический адрес отправителя, IP-адрес отправителя, физический адрес получателя и IP-адрес получателя. ARP-протокол обеспечивает связь между этими двумя адресами. Существует четыре типа ARP-сообщений: ARP-запрос (ARP request), ARP-ответ (ARP reply), RARP-запрос (RARP-request) и RARP-ответ (RARP-reply). Локальный хост при помощи ARP-запроса запрашивает физический адрес хоста-получателя. Ответ (физический адрес хоста-получателя) приходит в виде ARP-ответа. Хост-получатель, вместе с ответом, шлет также RARP-запрос, адресованный отправителю, для того, чтобы проверить его IP адрес. После проверки IP адреса отправителя, начинается передача пакетов данных.

Перед тем, как создать подключение к какому-либо устройству в сети, IP-протокол проверяет свой ARP-кеш, чтобы выяснить, не зарегистрирована ли в нём уже нужная для подключения информация о хосте-получателе. Если такой записи в ARP-кеше нет, то выполняется широковещательный ARP-запрос. Этот запрос для устройств в сети имеет следующий смысл: "Кто-нибудь знает физический адрес устройства, обладающего следующим IP-адресом?" Когда получатель примет этот пакет, то должен будет ответить: "Да, это мой IP-адрес. Мой физический адрес следующий: …" После этого отправитель обновит свой ARP-кеш, и будет способен передать информацию получателю. Ниже приведён пример ARP-запроса и ARP-ответа. *<см. внизу страницы>*

Записи в ARP-кеше могут быть статическими и динамическими. Пример, данный выше, описывает динамическую запись кеша. Хост-отправитель автоматически послал запрос получателю, не уведомляя при этом пользователя. Записи в ARP-кеш можно добавлять вручную, создавая статические записи кеша. Это можно сделать при помощи команды:

```
arp -s <IP адрес> <MAC адрес>
```

После того, как IP-адрес прошёл процедуру разрешения адреса, он остаётся в кеше в течение 2-х минут. Если в течение этих двух минут произошла повторная передача данных по этому адресу, то время хранения записи в кеше продлевается ещё на 2 минуты. Эта процедура может повторяться до тех пор, пока запись в кеше просуществует до 10 минут. После этого запись будет удалена из кеша и будет отправлен повторный ARP-запрос.

Вариации ARP-протокола

ARP изначально был разработан не только для IP протокола, но в настоящее время в основном используется для сопоставления IP- и MAC-адресов.

ARP также можно использовать для разрешения MAC-адресов для различных адресов протоколов 3-го уровня (Layer 3 protocols addresses). ARP был адаптирован также для разрешения других видов адресов 2-го уровня (Layer 2 addresses); например, ATMARP используется для разрешения ATM NSAP адресов в Classical IP over ATM протоколе.

Inverse ARP

Inverse Address Resolution Protocol, **Inverse ARP** или **InARP** - протокол для получения Layer 3 адреса (например IP адресов) других рабочих станций по Layer 2 адресам (например DLCI в Frame Relay сетях). В основном используется во Frame Relay и ATM сетях.

Сравнение ARP и InARP

ARP переводит Layer 3 адреса в Layer 2 адреса, в то же время InARP можно рассматривать как его инверсию. InARP реализовано как расширение ARP. Форматы пакетов этих протоколов одни и те же, отличаются лишь коды операций и заполняемые поля.

Reverse ARP (RARP), как и InARP, также переводит Layer 2 адреса в Layer 3 адреса. Но RARP используется для получения Layer 3 адресов самих станций отправителей, в то время как в InARP-протоколе отправитель

знает свои Layer 2 и Layer 3 адреса, и запрашивает Layer 3 адрес другой станции. От RARP отказались в пользу BOOTP который был в свою очередь заменён DHCP.

Структура пакета

Ниже проиллюстрирована структура пакета, используемого в ARP-запросах и ответах. В сетях Ethernet эти пакеты используют EtherType 0x0806, и рассылаются широковещательно MAC-адрес - FF:FF:FF:FF:FF:FF. Отметим, что в структуре пакета, показанной ниже в качестве SHA, SPA, THA, & TPA условно используются 32-битные слова — реальная длина определяется физическим устройством и протоколом.

	Bits 0 - 7	8 - 15	16 - 31
	Hardware type (HTYPE)		Protocol type (PTYPE)
2	Hardware length (HLEN)	Protocol length (PLEN)	Operation (OPER)
4	Sender hardware address (SHA)		
?	Sender protocol address (SPA)		
?	Target hardware address (THA)		
?	Target protocol address (TPA)		

Hardware type (HTYPE)

Каждый транспортный протокол передачи данных имеет свой номер, который хранится в этом поле. Например, Ethernet имеет номер 1.

Protocol type (PTYPE)

Код протокола. Например, для IPv4 будет записано 0x0800.

Hardware length (HLEN)

длина физического адреса в байтах. Ethernet адреса имеют длину 6 байт.

Protocol length (PLEN)

длина логического адреса в байтах. IPv4 адреса имеют длину 4 байта.

Operation

Код операции отправителя: 1 в случае запроса и 2 в случае ответа.

Sender hardware address (SHA)

Физический адрес отправителя.

Sender protocol address (SPA)

Логический адрес отправителя.

Target hardware address (THA)

Физический адрес получателя. Поле пусто при запросе.

Target protocol address (TPA)

Логический адрес получателя.

Пример запроса

Если хост с IPv4 адресом 10.10.10.123 и MAC адресом 00:09:58:D8:11:22 хочет послать пакет другому хосту с адресом 10.10.10.140, но не знает его MAC адрес, то он должен послать ARP запрос для разрешения адреса.

Пакет, изображённый ниже, изображает широковещательный запрос. Если хост с IP 10.10.10.140 присутствует в сети и доступен, то он получает этот ARP-запрос и возвращает ответ.

	Bits 0 - 7	8 - 15	16 - 31
	Hardware type = 1		Protocol type = 0x0800
2	Hardware length = 6	Protocol length = 4	Operation = 1
4	SHA (first 32 bits) = 0x000958D8		
6	SHA (last 16 bits) = 0x1122		SPA (first 16 bits) = 0x0A0A
28	SPA (last 16 bits) = 0x0A7B		THA (first 16 bits) = 0x0000

60	THA (last 32 bits) = 0x00000000
92	TPA = 0x0A0A0A8C

Пример ответа

По сценарию, описанному выше, если хост с адресом 10.10.10.140 имеет MAC адрес 00:09:58:D8:33:AA, то он пошлет в ответ пакет, проиллюстрированный ниже. Заметим, что блоки адресов отправителя и получателя теперь поменяли значения (отправитель ответа теперь получатель запроса; получатель ответа - отправитель запроса). Кроме того, хост 10.10.10.140 заполнил свой MAC адрес в поле физического адреса отправителя.

Любой хост в той же сети, что и отправитель с получателем, тоже получат запрос (так как он широковещательный) и таким образом добавят в свой кеш информацию об отправителе. ARP-ответ направлен только источнику ARP-запроса, поэтому ARP-ответ не доступен другим хостам в сети.

	Bits 0 - 7	**8 - 15**	**16 - 31**
	Hardware type = 1		Protocol type = 0x0800
2	Hardware length = 6	Protocol length = 4	Operation = 2
4	SHA (first 32 bits) = 0x000958D8		
6	SHA (last 16 bits) = 0x33AA		SPA (first 16 bits) = 0x0A0A
28	SPA (last 16 bits) = 0x0A8C		THA (first 16 bits) = 0x0009
60	THA (last 32 bits) = 0x58D81122		
92	TPA = 0x0A0A0A7B		

ARP Оповещение

- **Замечание:** Длинна полей SHA,SPA,THA,TPA зависит от параметров Hardware length и Protocol length соответственно, в данном случае таблицы не совсем верны, при посылке запроса к станции с неизвестным MAC адресом в поле назначения нужно указывать широковещательный адрес: FF FF FF FF FF FF (-kipish-)

ARP оповещение (ARP Announcement) - это пакет (обычно ARP запрос [1]) содержащий корректную SHA и SPA хоста-отправителя, с TPA равной SPA. Это не разрешающий запрос, а запрос на обновление ARP-кеша других хостов, получающих пакет.

Большинство операционных систем посылают такой пакет при включении хоста в сеть, это позволяет предотвратить ряд проблем. Например при смене сетевой карты (когда необходимо обновить связь между IP и MAC адресами), такой запрос исправит записи в ARP-кеше других хостов в сети.

ARP оповещения также используются для 'защиты' IP адресов в RFC3927 (Zeroconf) протоколе.

7. Понятие маски подсети, ее назначение. Безклассовая модель представления сетевых адресов.

8. Протоколы транспортного уровня (TCP, UDP).

TCP (англ. *Transmission Control Protocol* — протокол управления передачей) — один из основных сетевых протоколов Internet, предназначенный для управления передачей данных в сетях и подсетях TCP/IP.

Выполняет функции протокола транспортного уровня упрощённой модели OSI. IP-идентификатор — 6.

TCP — это транспортный механизм, предоставляющий поток данных, с предварительной установкой соединения, за счёт этого дающий уверенность в безошибочности получаемых данных, осуществляет повторный запрос данных в случае потери пакетов и устраняет дублирование при получении двух копий одного пакета (см. также T/TCP). В отличие от UDP, TCP гарантирует, что приложение получит данные точно в такой же последовательности, в какой они были отправлены, и без потерь.

Формат TCP-сегмента

Формат TCP-сегмента				
Бит	**0 — 3**	**4 — 7**	**8 — 15**	**16 — 31**

0	Порт источника			Порт назначения
32	Номер последовательности			
64	Номер подтверждения			
96	Смещение данных	Зарезервировано	Флаги	Окно
128	Контрольная сумма			Указатель важности
160	Опции (необязательное)			
160/192 +	Данные			

Порт источника идентифицирует порт, с которого отправлен пакет.
Порт назначения идентифицирует порт, на который отправлен пакет
Номер последовательности выполняет две задачи:

1. Если установлен флаг SYN, то это начальное значение номера последовательности и первый байт данных — это номер последовательности плюс 1.
2. В противном случае, если SYN не установлен, первый байт данных — номер последовательности

Номер подтверждения

Если установлен флаг ACK, то это поле содержит номер последовательности, ожидаемый отправителем в следующий раз. Помечает этот сегмент как подтверждение получения.

Смещение данных

Это поле определяет размер заголовка пакета TCP в 32-битных словах. Минимальный размер составляет 5 слов, а максимальный — 15, что составляет 20 и 60 байт соответственно. Смещение считается от начала заголовка TCP.

Зарезервировано

Зарезервировано(4 бита) для будущего использования и должны устанавливаться в ноль.

Флаги (управляющие биты)

Это поле содержит 8 битовых флагов:

- **CWR** (Congestion Window Reduced) — Поле «Окно перегрузки уменьшено» — флаг установлен отправителем, чтоб указать, что получен пакет с установленным флагом ECE (RFC 3168)
- **ECE** (ECN-Echo) — Поле «Эхо ECN» — указывает, что данный хост способен на ECN (явное уведомление перегрузки) и для указания отправителю о перегрузках в сети (RFC 3168)
- **URG** — Поле *Указатель важности* значимо (англ. *Urgent pointer field is significant*)
- **ACK** — Поле *Номер подтверждения* значимо (англ. *Acknowledgement field is significant*)
- **PSH** — (англ. *Push function*) инструктирует получателя протолкнуть данные, накопившиеся в приемном буфере, в приложение пользователя
- **RST** — Оборвать соединения, сбросить буфер (очистка буфера) (англ. *Reset the connection*)
- **SYN** — Синхронизация номеров последовательности (англ. *Synchronize sequence numbers*)
- **FIN** (англ. *final*, бит) — флаг, будучи установлен, указывает на завершение соединения (англ. *FIN bit used for connection termination*).

Контрольная сумма

Поле контрольной суммы — это 16-битное дополненение суммы всех 16-битных слов заголовка и текста. Если сегмент содержит нечетное число октетов в заголовке /или тексте, последние октеты дополняются справа 8 нулями для выравнивания по 16-битовой границе. Биты заполнения (0) не передаются в сегменте и служат только для расчета контрольной суммы. При расчете контрольной суммы значение самого поля контрольной суммы принимается равным 0.

Указатель важности

16-битовое значение положительного смещения от порядкового номера в данном сегменте. Это поле указывает порядковый номер октета, с которого начинаются важные (urgent) данные. Поле принимается во внимание только для пакетов с установленным флагом URG.

Механизм действия протокола

Состояния сеанса TCP

Состояния сеанса TCP	
CLOSED	Начальное состояние узла. Фактически фиктивное
LISTEN	Сервер ожидает запросов установления соединения от клиента
SYN-SENT	Клиент отправил запрос серверу на установление соединения и ожидает ответа
SYN-RECIEVED	Сервер получил запрос на соединение, отправил ответный запрос и ожидает подтверждения
ESTABLISHED	Соединение установлено, идёт передача данных
FIN-WAIT-1	Одна из сторон (назовём её узел-1) завершает соединение, отправив сегмент с флагом FIN
CLOSE-WAIT	Другая сторона (узел-2) переходит в это состояние, отправив, в свою очередь сегмент ACK и продолжает одностороннюю передачу
FIN-WAIT-2	Узел-1 получает ACK, продолжает чтение и ждёт получения сегмента с флагом FIN
LAST-ACK	Узел-2 заканчивает передачу и отправляет сегмент с флагом FIN
TIME-WAIT	Узел-1 получил сегмент с флагом FIN, отправил сегмент с флагом ACK и ждёт 2*MSL секунд, перед окончательным разрушением канала
CLOSING	Состояние закрытия соединения (фиктивное?)

Установка соединения

Процесс начала сеанса TCP называется "тройным рукопожатием". Клиент, который намеревается установить соединение, посылает серверу сегмент с номером последовательности и флагом SYN. Сервер получает сегмент, запоминает номер последовательности и пытается создать сокет (буфера и управляющие структуры памяти) для обслуживания нового клиента. В случае успеха сервер посылает клиенту сегмент с номером последовательности и флагами SYN и ACK, и переходит в состояние SYN-RECEIVED. В случает неудачи сервер посылает клиенту сегмент с флагом RST.

Если клиент получает сегмент с флагом SYN, то он запоминает номер последовательности и посылает сегмент с флагом ACK, если он одновременно получает и флаг ACK (что обычно и происходит), то он переходит в состояние ESTABLISHED. Если клиент получает сегмент с флагом RST, то он прекращает попытки соединиться, в противном случае клиент повторяет процесс установки соединения.

Если клиент не получает ответа в течении 10 секунд, то он повторяет процесс соединения заново.

Если сервер в состоянии SYN-RECEIVED получает сегмент с флагом ACK, то он переходит в состояние ESTABLISHED. В противном случае после таймаута он закрывает сокет и переходит в состояние CLOSED.

Процесс называется "тройным рукопожатием", поскольку в идеале возможен процесс установления соединения с использованием 4 сегментов (SYN в сторону сервера, ACK в сторону клиента, SYN в сторону клиента, ACK в сторону сервера), но для экономии времени используется 3 сегмента.

Передача данных

При обмене данными приемник использует номер последовательности, содержащийся в получаемых сегментах, для восстановления их исходного порядка. Приемник уведомляет передающую сторону о номере последовательности, до которой он успешно получил данные, включая его в поле "номер подтверждения". Все получаемые данные, относящиеся к промежутку подтвержденных последовательностей, игнорируются. Если полученный сегмент содержит номер последовательности больший, чем ожидаемый, то данные из сегмента буферизируется, но номер подтвержденной последовательности не изменяется. Если в последствии будет принят сегмент, относящийся к ожидаемому номеру последовательности, то порядок данных будет автоматически восстановлен исходя из номеров последовательностей в сегментах.

Для того, чтобы передающая сторона не отправляла данные интенсивнее, чем их может обработать приемник, TCP содержит средства управления потоком. Для этого используется поле "окно". В сегментах, направляемых от приемника передающей стороне в поле "окно" указывается текущий размер приемного буфера. Передающая сторона сохраняет размер окна и отправляет данных не более, чем указал приемник. Если приемник указал нулевой размер окна, то передача данных в направлении этого узла не происходит, до тех пор пока приемник не сообщит о большем размере окна.

В некоторых случаях передающее приложение может явно затребовать протолкнуть данные до некоторой последовательности принимающему приложению, не буферизируя их. Для этого используется флаг PSH. Если в полученном сегменте обнаруживается флаг PSH, то реализация TCP отдает все буферизированные на текущий момент данные принимающему приложению. "Проталкивание" используется, например, в интерактивных приложениях. В сетевых терминалах нет смысла ожидать ввода пользователя после того, как он закончил набирать команду. Поэтому последний сегмент, содержащий команду, обязан содержать флаг PSH, чтобы приложение на принимающей стороне смогло начать её выполнение.

UDP (англ. *User Datagram Protocol* — протокол пользовательских датаграмм) — это транспортный протокол для передачи данных в сетях IP. Он является одним из самых простых протоколов транспортного уровня модели OSI. Его IP-идентификатор — 17.

В отличие от TCP, UDP не гарантирует доставку пакета, поэтому аббревиатуру иногда расшифровывают как «*Unreliable Datagram Protocol*» (протокол ненадёжных датаграмм). Это позволяет ему гораздо быстрее и эффективнее доставлять данные для приложений, которым требуется большая пропускная способность линий связи, либо требуется малое время доставки данных.

Порты

Для взаимодействия сетевых приложений протокол UDP использует 16-ти битные порты, которые могут принимать значения от 0 до 65535. Порт 0 является зарезервированным, но может использоваться как порт источника, если приложение не ожидает ответных данных.

Порты с 1 по 1023 являются системными и фиксированными, во многих ОС привязка к ним требует повышенных привилегий приложения.

Порты с 1024 по 49151 — зарегистрированные.

Порты с 49152 по 65535 — свободно используемые и временные. Используются клиентскими приложениями для связи с серверами.

Формат сегмента

Заголовок UDP содержит 4 поля.

Поле «порт отправителя» (16 бит) определяет процесс на хосте отправителя, пославший пакет. В случае, если процесс-отправитель не ожидает от получателя никаких ответных данных, это поле может быть установлено в «0».

Поле «порт получателя» (16 бит) определяет процесс на хосте получателя, которому предназначен данный пакет.

Поле «Длина пакета» (16 бит) содержит суммарный размер UDP-пакета в октетах. Минимально возможное значение этого поля равно 8 (т.к. 8 октетов занимает сам заголовок пакета).

Поле «контрольная сумма» имеет длину 16 бит.

Формат UDP-сегмента

	Биты 0—15	**16—31**
	Порт отправителя	Порт получателя
2	Длина пакета	Контрольная сумма
4	Данные	

Очевидно, что максимальная длина UDP-пакета составляет 2^{16}-1 = 65535 октетов. Вычитая из этой длины размер заголовка (8 октетов), получаем максимальный размер данных, которые могут быть переданы в одном пакете — 65527 октетов.

Недостаточная надёжность протокола может выражаться как в потере отдельных пакетов, так и в их дублировании. UDP используется при передаче потокового видео, игр реального времени, а также некоторых других типов данных.

Если приложению требуется большая надёжность, то используется протокол TCP или SCTP.

Использование

UDP используется в следующих протоколах:

- DNS
- RTP и RTCP
- TFTP
- SNTP

- NTP

9. Протокол ICMP. Модель, основные команды, безопасность, производительность.

ICMP (англ. Internet Control Message Protocol — межсетевой протокол управляющих сообщений) — сетевой протокол, входящий в стек протоколов TCP/IP. В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных. Также на ICMP возлагаются некоторые сервисные функции.

Протокол ICMP описан в RFC 792 (с дополнениями в RFC 950) и является стандартом Интернета (входит в стандарт STD 5 вместе с IP). Хотя формально ICMP использует IP (ICMP пакеты инкапсулируются в IP пакеты), он является неотъемлемой частью IP и обязателен при реализации стека TCP/IP. Текущая версия ICMP для IPv4 называется ICMPv4. В IPv6 существует аналогичный протокол ICMPv6.

Протокол ICMP не делает протокол IP средством надёжной доставки сообщений. Для этих целей существует TCP.

ICMP сообщения (тип 12) генерируются при нахождении ошибок в заголовке IP пакета (за исключением самих ICMP пакетов, дабы не привести к бесконечно растущему потоку ICMP сообщений об ICMP сообщениях).

ICMP сообщения (тип 3) генерируются маршрутизатором при отсутствии маршрута к адресату.

Утилита ping, служащая для проверки возможности доставки IP пакетов использует ICMP сообщения с типом 8 (эхо-запрос) и 0 (эхо-ответ).

Утилита traceroute, отображающая путь следования IP пакетов, использует ICMP сообщения с типом 11.

ICMP сообщения с типом 5 используются маршрутизаторами для обновления записей в таблице маршрутизации отправителя.

ICMP сообщения с типом 4 используются получателем (или промежуточным маршрутизатором) для управления скоростью отправки сообщений отправителем.

Формат ICMP-пакета

Формат ICMP-пакета

ит	**0 —7**	**8— 15**	**16—31**
	Тип	Код	Контрольная сумма
2	Содержание сообщения (зависит от значений полей «Код» и «Тип»)		

Типы ICMP пакетов (неполный список)

- **0** — Эхо-ответ
- **1** — Зарезервировано
- **2** — Зарезервировано
- **3** — Адресат недоступен

код 0 — сеть недостижима;
код 1 — хост недостижим;
код 2 — протокол недостижим;
код 3 — порт недостижим;
код 4 — необходима фрагментация, но установлен флаг ее запрета(DF);
код 5 — неверный маршрут от источника;
код 6 — Сеть назначения неизвестна;
код 7 — Хост назначения неизвестен;
код 8 — Хост источник изолированн;
код 9 — Сеть административно запрещена;
код 10 — Хост административно запрещен;
код 11 — Сеть недоступна для TOS;
код 12 — Хост недоступен для TOS;
код 13 — Коммуникации административно запрещены;

- **4** — Сдерживание источника (отключение источника при переполнении очереди)
- **5** — Перенаправление

код 0 — перенаправление пакетов в сеть;
Код 1 — перенаправление пакетов к хосту;
Код 2 — перенаправление для каждого типа обслуживания(TOS);
Код 3 — перенаправление пакета к хосту для каждого типа обслуживания;

- **6** — Альтернативный адрес хоста
- **7** — Зарезервировано
- **8** — Эхо-запрос
- **9** — Объявление маршрутизатора
- **10** — Запрос маршрутизатора

- **11** — Превышение временно́го интервала (для дейтаграммы время жизни истекло)

тип 0 — время жизни пакета истекло при транспортировке тип 1 — время жизни пакета истекло при дефрагментации

- **12** — Неверный параметр (проблема с параметрами дейтаграммы: ошибка в IP-заголовке или отсутствует необходимая опция)
- **13** — Запрос метки времени
- **14** — Ответ с меткой времени
- **15** — Информационный запрос
- **16** — Информационный ответ
- **17** — Запрос адресной маски
- **18** — Отклик на запрос адресной маски

10. Протокол POP. Модель, основные команды, безопасность, производительность.

POP3 (англ. *Post Office Protocol Version 3* — протокол почтового отделения, версия 3) используется почтовым клиентом для получения сообщений электронной почты с сервера. Обычно используется в паре с протоколом SMTP.

Предыдущие версии протокола (POP, POP2) устарели.

Стандарт протокола POP3 определён в RFC 1939. Расширения и методы авторизации определены в RFC 2195, RFC 2449, RFC 1734, RFC 2222, RFC 3206, RFC 2595.

Существуют реализации POP3-серверов, поддерживающие TLS и SSL.

Альтернативным протоколом для сбора сообщений с почтового сервера является IMAP.

Состояния сеанса

В протоколе POP3 предусмотрено 3 состояния сеанса:

Авторизация

Клиент проходит процедуру Аутентификации

Транзакция

Клиент получает информацию о состоянии почтового ящика, принимает и удаляет почту

Обновление

Сервер удаляет выбранные письма и закрывает соединение

Команды протокола

APOP [имя] [digest]

Команда служит для передачи серверу имени пользователя и зашифрованного пароля(digest)

Аргументы

[имя] - строка, указывающая имя почтового ящика. [digest]- временная метка, зашифрованная паролем пользователя по алгоритму MD5. В случае поддержки этой команды временная метка получается при соединении с сервером:

+OK POP3 server ready <**1896.698370952@meshach.smallorg.org**>

Ограничения

Её поддержка не является обязательной

Возможные ответы

- +OK maildrop has n message
- -ERR password suplied for [имя] is incorrect

DELE [сообщение]

Сервер помечает указанное сообщение для удаления. Сообщения, помеченные на удаление, реально удаляются только после закрытия транзакции (закрытие транзакций происходит обычно после посыла команды QUIT, кроме этого, например, на серверах закрытие транзакций может происходить по истечению определенного времени, установленного сервером).

Аргументы

[сообщение] - номер сообщения.

Ограничения

Доступна после успешной идентификации

Возможные ответы

- +OK message deleted
- -ERR no such message

LIST [сообщение]

Если был передан аргумент, то сервер выдаёт информацию о указанном сообщении. Если аргумент не был передан, то сервер выдаёт информацию о всех сообщениях, находящихся в почтовом ящике. Сообщения, помеченные для удаления не перечисляются.

Аргументы

[сообщение]-номер сообщения (необязательный аргумент)

Ограничения

Доступна после успешной идентификации

Возможные ответы

- +OK scan listing follows
- -ERR no such message

NOOP

Сервер ничего не делает, всегда отвечает положительно

Аргументы

Нет.

Ограничения

Нет.

Возможные ответы

- +OK

PASS [пароль]

Передаёт серверу пароль почтового ящика

Аргументы

[пароль] - пароль для почтового ящика.

Ограничения

Работает после успешной передачи имени почтового ящика.

Возможные ответы

- +OK maildrop locked and ready
- -ERR invalid password
- -ERR unable to lock maildrop

RETR [сообщение]

Сервер передаёт сообщение с указанным номером

Аргументы

[сообщение] - номер сообщения

Ограничения

Доступна после успешной идентификации

Возможные ответы

- +OK message follows
- -ERR no such message

RSET

Этой командой производится откат транзакций внутри сессии. Например, если пользователь случайно пометил на удаление какие-либо сообщения, он может убрать эти пометки, отправив эту команду

Аргументы

Нет.

Ограничения

Доступна после и до успешной идентификации

Возможные ответы

- +OK

STAT

Сервер возвращает количество сообщений в почтовом ящике плюс размер, занимаемыми этими сообщениями на почтовом ящике

Аргументы

Нет

Ограничения

Доступна после успешной идентификации

Возможные ответы

- +OK a b

TOP [сообщение] [количество строк]

Сервер возвращает указанное количество строк после заголовка

Аргументы

[сообщение] - номер сообщения

[количество строк] - сколько строк нужно вывести

Ограничения

Доступна после успешной идентификации

Возможные ответы

- +OK n octets
- -ERR no such message

USER [имя]

Передаёт серверу имя пользователя

Аргументы

[имя] - строка, указывающая имя почтового ящика.

Ограничения

Нет.

Возможные ответы

- +OK name is a valid mailbox
- -ERR never heard of mailbox name

QUIT

Аргументы

Нет.

Ограничения

Нет.

Возможные ответы

- +OK

Пример сессии

Место POP3 в передаче почты (e-mail)

Это пример сессии с поддержкой зашифрованных паролей (APOP, RFC 1939):

```
S: <Сервер ожидает входящих соединений на порту 110>
C: <подключается к серверу>
S:    +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>
C:    APOP mrose c4c9334bac560ecc979e58001b3e22fb
S:    +OK mrose's maildrop has 2 messages (320 octets)
C:    STAT
S:    +OK 2 320
C:    LIST
S:    +OK 2 messages (320 octets)
S:    1 120
S:    2 200
S:    .
C:    RETR 1
S:    +OK 120 octets
S:    <сервер передает сообщение 1>
S:    .
C:    DELE 1
S:    +OK message 1 deleted
C:    RETR 2
S:    +OK 200 octets
S:    <сервер передает сообщение 2>
S:    .
C:    DELE 2
S:    +OK message 2 deleted
C:    QUIT
S:    +OK dewey POP3 server signing off (maildrop empty)
C:  <закрывает соединение>
S:  <продолждает ждать входящие соединения>
```

Вариант начала сессии, при котором пароль передается открытым текстом.:

```
C:    USER mrose
S     +OK User accepted
C:    PASS mrosepass
S     +OK Pass accepted
```

11. Протокол SMTP. Модель, основные команды, безопасность, производительность.

SMTP (англ. *Simple Mail Transfer Protocol* — простой протокол передачи почты) — это сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP.

ESMTP (англ. *Extended SMTP*) — масштабируемое расширение протокола SMTP. В настоящее время под «протоколом SMTP», как правило, подразумевают ESMTP и его расширения.

Обзор протокола

SMTP используется для отправки почты от пользователей к серверам и между серверами для дальнейшей пересылки к получателю. Для приёма почты почтовый клиент должен использовать протоколы POP3 или IMAP.

Чтобы доставить сообщение до адресата, необходимо переслать его почтовому серверу домена, в котором находится адресат. Для этого обычно используется запись типа MX (англ. *Mail eXchange* — обмен почтой) системы DNS. Если MX запись отсутствует, то для тех же целей может быть использована запись типа A. Некоторые современные реализации SMTP-серверов (например, Exim[1]) для определения сервера, обслуживающего почту в домене адресата, также могут задействовать SRV-запись (RFC 2782).

Широкое распространение SMTP получил в начале 1980-х годов. До него использовался протокол UUCP, который требовал от отправителя знания полного маршрута до получателя и явного указания этого маршрута в адресе получателя, либо наличия прямого коммутируемого или постоянного соединения между компьютерами отправителя и получателя.

Sendmail был одним из первых (если не первым) агентом отправки сообщений, который начал работать с SMTP. В настоящее время протокол SMTP является стандартным для электронной почты и его используют все клиенты и серверы.

Протокол был разработан для передачи только текста в кодировке ASCII, кроме того, первые спецификации требовали обнуления старшего бита каждого передаваемого байта. Это не даёт возможности отсылать текст на национальных языках (например, кириллице), а также отправлять двоичные файлы (такие как изображения, видеофайлы, программы или архивы). Для снятия этого ограничения был разработан стандарт MIME, который описывает способ преобразования двоичных файлов в текстовые. В настоящее время большинство серверов поддерживают 8BITMIME, позволяющий отправлять двоичные файлы так же просто, как текст.

Сервер SMTP — это конечный автомат с внутренним состоянием. Клиент передает на сервер строку `команда<пробел>параметры<перевод строки>`. Сервер отвечает на каждую команду строкой, содержащей код ответа и текстовое сообщение, отделенное пробелом. Код ответа — число от 100 до 999, представленное в виде строки, трактующийся следующим образом:

- 2XX — команда успешно выполнена
- 3XX — ожидаются дополнительные данные от клиента
- 4XX — временная ошибка, клиент должен произвести следующую попытку через некоторое время
- 5XX — неустранимая ошибка

Текстовая часть ответа носит справочный характер и предназначен для человека, а не программы.

ESMTP — расширяемый протокол, в отличии от SMTP. При установлении соединения сервер объявляет о наборе поддерживаемых расширений (в качестве ответа на команду **EHLO**). Соответствующие расширения могут быть использованы клиентом при работе. Необходимо помнить, что если сессия начинается с команды **HELO** (используемой в «классическом» SMTP, RFC 821), то список расширений выводиться не будет.

Безопасность SMTP и спам

Изначально SMTP не поддерживал единой схемы авторизации. В результате этого спам стал практически неразрешимой проблемой, так как было невозможно определить, кто на самом деле является отправителем сообщения — фактически можно отправить письмо от имени любого человека. В настоящее время производятся попытки решить эту проблему при помощи спецификаций SPF, Sender ID, Yahoo Domain Keys. Единой спецификации на настоящий момент не существует.

Пример простейшей SMTP-сессии

C: — клиент, S: — сервер

```
S: (ожидает соединения)
C: (Подключается к порту 25 сервера)
S:220 mail.company.tld ESMTP CommuniGate Pro 5.1.4i is glad to see you!
C:HELO
S:250 domain name should be qualified
C:MAIL FROM: <someusername@somecompany.ru>
S:250 someusername@somecompany.ru sender accepted
C:RCPT TO:<user1@company.tld>
S:250 user1@company.tld ok
C:RCPT TO: <user2@company.tld >
S:550 user2@company.tld  unknown user account
C:DATA
S:354 Enter mail, end with "." on a line by itself
C:Hi!
C:.
S:250 769947 message accepted for delivery
C:QUIT
S:221 mail.company.tld CommuniGate Pro SMTP closing connection
S: (закрывает соединение)
```

В результате такой сессии письмо будет доставлено адресату user1@company.tld, но не будет доставлено адресату user2@company.tld, потому что такого адреса не существует.

Команды SMTP

- `HELO <SP> <string><CRLF>` — Идентифицирует SMTP-сервер отправителя, открывает сеанс

Пример

```
HELO user.example.net
250 server.example.com Hello user.example.net [192.168.1.1] pleased to meet you
```

- `QUIT<CRLF>` — Завершает SMTP-сеанс.

Пример

```
QUIT
221 2.0.0 server.example.com closing connection
```

- `MAIL <SP> FROM:<reverse-path> <CRLF>` — Задает адрес отправителя. Адрес следует указывать в угловых скобках. Некоторые серверы могут проигнорировать то, что им передают адрес без угловых скобок, но те серверы, что неукоснительно следуют описанию RFC, отклонят такой адрес.

Пример

```
MAIL FROM: <USER@EXAMPLE.NET>
250 2.1.0 USER@EXAMPLE.NET... Sender ok
```

- `RCPT <SP> TO:<forward-path> <CRLF>` — Задает адрес получателя. Адрес следует указывать в угловых скобках. Некоторые серверы могут проигнорировать то, что им передают адрес без угловых скобок, но те серверы, что неукоснительно следуют описанию RFC, отклонят такой адрес.

Пример

```
RCPT TO: <USER2@EXAMPLE.COM>
250 2.1.5 USER2@EXAMPLE.COM... Recipient ok
```

- `DATA <CRLF>` — Указывает на начало сообщения. Для окончания сообщения указывается `<CRLF>.<CRLF>`.

Пример:

```
DATA
354 Enter mail, end with "." on a line by itself
this is a test message.
.
250 2.0.0 l3PDY91f000484 Message accepted for delivery
```

- `VRFY <SP> <string><CRLF>` — проверяет существование получателя.

Пример (пояснение чуть ниже):

```
VRFY
252 2.5.2 Cannot VRFY user; try RCPT to attempt delivery (or try finger)
```

- `EXPN <SP> <string><CRLF>` — показывает список адресов для списка рассылки.

Пример (пояснения чуть ниже):

```
EXPN
502 5.7.0 Sorry, we do not allow this operation
```

- `NOOP<CRLF>` — пустая операция

Пример:

```
NOOP
250 2.0.0 OK
```

- `TURN<CRLF>` — сервер и клиент меняются ролями после ответа сервера 200 OK

Пример: (команда не реализована)

```
TURN
502 5.5.1 Command not implemented: "TURN"
```

- `RSET<CRLF>` — сброс сессии в исходное состояние

Пример:

```
RSET
250 2.0.0 Reset state
```

- `HELP<CRLF>` — информация о поддерживаемых командах. Некоторые сервера поддерживают справку по отдельным командам, например, `HELP MAIL` (sendmail), некоторые выводят по этой команде лишь список доступных команд без пояснения (Microsoft Exchange Server).

12. Протокол FTP. Модель, основные команды, безопасность, производительность.

FTP (англ. *File Transfer Protocol* — протокол передачи файлов) — протокол, предназначенный для передачи файлов в компьютерных сетях. FTP позволяет подключаться к серверам FTP, просматривать содержимое

каталогов и загружать файлы с сервера или на сервер; кроме того, возможен режим передачи файлов между серверами (см. FXP).

FTP является одним из старейших прикладных протоколов, появившимся задолго до HTTP, в 1971 году. До начала 90-х годов на долю FTP приходилось около половины трафика в сети Интернет[источник?]. Он и сегодня широко используется для распространения ПО и доступа к удалённым хостам.

Протокол не шифруется, при аутентификации передаёт логин и пароль открытым текстом. Если злоумышленник находится в одном сегменте сети с пользователем FTP, то, используя сниффер, он может перехватить логин и пароль пользователя, или, при наличии специального ПО, получать передаваемые по FTP файлы без авторизации. Чтобы предотвратить перехват трафика, необходимо использовать протокол шифрования данных SSL, который поддерживается многими современными FTP-серверами и некоторыми FTP-клиентами.

Процесс нешифрованной авторизации проходит в несколько этапов (символы \r\n означают перевод строки):

```
* Установка TCP-соединения с сервером (обычно на 21 порт)
* Посылка команды USER логин\r\n
* Посылка команды PASS пароль\r\n
```

Если к серверу разрешён анонимный доступ, то можно авторизоваться так:

```
* USER anonymous\r\n
* PASS someone@email\r\n
```

После успешной авторизации можно посылать на сервер другие команды.

На многих FTP-серверах существует каталог (под названием incoming, upload и т. п.), открытый на запись и предназначенный для закачки файлов на сервер. Это позволяет пользователям наполнять сервер свежими данными.

Изначально протокол предполагал встречное TCP-соединение от сервера к клиенту для передачи файла или содержимого каталога. Это делало невозможным общение с сервером, если клиент находится за IP NAT, кроме того, часто запрос соединения к клиенту блокируется файерволом. Чтобы этого избежать, было разработано расширение протокола FTP passive mode, когда соединение для передачи данных тоже происходит от клиента к серверу. Кроме того, этой проблемы можно избежать, если использовать прокси-сервер.

13.Протокол SNMP. Модель, основные команды, безопасность, производительность.

SNMP (англ. *Simple Network Management Protocol* — простой протокол управления сетью) — это протокол управления сетями связи на основе архитектуры TCP/IP.

На основе концепции TMN в 1980—1990 гг. различными органами стандартизации был выработан ряд протоколов управления сетями передачи данных с различным спектром реализации функций TMN. К одному из типов таких протоколов управления относится SNMP.

Также это технология, призванная обеспечить управление и контроль за устройствами и приложениями в сети связи путём обмена управляющей информацией между агентами, располагающимися на сетевых устройствах, и менеджерами, расположенными на станциях управления. В настоящее время SNMP является базовым протоколом управления сети Internet. SNMP определяет сеть как совокупность сетевых управляющих станций и элементов сети (главные машины, шлюзы и маршрутизаторы, терминальные серверы), которые совместно обеспечивают административные связи между сетевыми управляющими станциями и сетевыми агентами.

Обзор и базовые концепции

Обычно при использовании SNMP присутствуют управляемые и управляющие системы. В состав управляемой системы входит компонент, называемый *агентом*, который отправляет отчеты управляющей системе. По существу SNMP агенты передают управленческую информацию на управляющие системы как переменные (такие как "свободная память", "имя системы", "количество работающих процессов").

Управляющая система может получить информацию через операции протокола **GET**, **GETNEXT** и **GETBULK**. Агент может самостоятельно без запроса отправить данные, используя операцию протокола **TRAP** или **INFORM**. Управляющие системы могут также отправлять конфигурационные обновления или контролирующие запросы, используя операцию **SET** для непосредственного управления системой. Операции конфигурирования и управления используются только тогда, когда нужны изменения в сетевой инфраструктуре. Операции мониторинга обычно выполняются на регулярной основе.

Переменные доступные через SNMP организованы в иерархии. Эти иерархии и другие метаданные (такие как тип и описание переменной) описываются *Базами Управляющей Информации* (англ. *Management Information Bases (MIBs)*).

Management Information Bases (MIBs)

Сам по себе SNMP не определяет, какая информация (какие переменные) управляемая система должна предоставлять. Зачастую SNMP использует расширяемую модель, где доступная информация определяется Базами Управляющей Информации (**MIB**). Базы Управляющей Информации описывают структуру управляющей информации устройств. Они используют иерархическое пространство имен содержащее уникальный идентификатор объекта (англ. *object identifier (**OID**)*). Грубо говоря, каждый уникальный идентификатор объекта идентифицирует переменную, которая может быть прочитана или установлена через SNMP. MIBы используют нотацию, определенную в ASN.1.

Иерархия MIB может быть изображена как дерево с безымянным корнем, уровни которого присвоены разными организациями. На самом высоком уровне MIB OIDы принадлежат разным организациям, пока на более

низком уровне OIDы выделяются ассоциированным организациям. Эта модель реализует управление через все слои сетевой модели OSI так как MIBы могут быть определены для всех подобных ситуаций и информации.

Управляемый объект - это одна из любого числа характеристик специфических для управляемого устройства. Управляемый объект включает в себя один или более экземпляров объекта (идентифицируемых по OID), которые на самом деле переменные.

Существует два типа управляемых объектов:

1. Скалярные объекты определяют единственный экземпляр объекта.
2. Табличные объекты определяют множественные, связанные экземпляры объектов которые группируются в таблицах MIB.

Примером управляемого объекта может быть **atInput**, который является скалярным объектом содержащим единственный экземпляр объекта, целое число, которое показывает общее количество входящих пакетов AppleTalk на сетевой интерфейс маршрутизатора.

Идентификатор объекта (OID) уникально идентифицирует управляемый объект в иерархии MIB.

14.NFS, RPC и XDR.

Network File System (**NFS**) — протокол сетевого доступа к файловым системам, первоначально разработан Sun Microsystems в 1984 году. Основан на протоколе вызова удалённых процедур (ONC RPC, Open Network Computing Remote Procedure Call, RFC 1057, RFC 1831). Позволяет подключать (монтировать) удалённые файловые системы через сеть, описан в RFC 1094, RFC 1813, и RFC 3530.

NFS абстрагирована от типов файловых систем как сервера, так и клиента, существует множество реализаций NFS-серверов и клиентов для различных операционных систем и аппаратных архитектур. В настоящее время (2007) используется наиболее зрелая версия NFS v.4 (RFC 3010), поддерживающая различные средства аутентификации (в частности, Kerberos и LIPKEY с использованием протокола RPCSEC_GSS) и списки контроля доступа (как POSIX, так и Windows-типов).

RPC (от англ. *Remote Procedure Call*) — технология, позволяющая компьютерным программам вызывать функции или процедуры в другом адресном пространстве (как правило, на удалённых компьютерах). Существуют множество технологий, обеспечивающих RPC:

- Sun RPC (RFC 1831)
- .Net Remoting
- XML RPC
- Java RMI
- Routix.RPC
- ZeroC ICE

Идея вызова удаленных процедур (Remote Procedure Call — RPC) состоит в расширении хорошо известного и понятного механизма передачи управления и данных внутри программы, выполняющейся на одной машине, на передачу управления и данных через сеть. Средства удаленного вызова процедур предназначены для облегчения организации распределенных вычислений. Наибольшая эффективность использования RPC достигается в тех приложениях, в которых существует интерактивная связь между удаленными компонентами с небольшим временем ответов и относительно малым количеством передаваемых данных. Такие приложения называются RPC-ориентированными.

Характерными чертами вызова локальных процедур являются:

- Асимметричность, то есть одна из взаимодействующих сторон является инициатором;
- Синхронность, то есть выполнение вызывающей процедуры приостанавливается с момента выдачи запроса и возобновляется только после возврата из вызываемой процедуры.

Реализация удаленных вызовов существенно сложнее реализации вызовов локальных процедур. Можно обозначить следующие проблемы и задачи, которые необходимо решить при реализации RPC:

- Т.к. вызывающая и вызываемая процедуры выполняются на разных машинах, то они имеют разные адресные пространства, и это создает проблемы при передаче параметров и результатов, особенно если машины не идентичны. Так как RPC не может рассчитывать на разделяемую память, то это означает, что параметры RPC не должны содержать указателей на ячейки нестековой памяти и что значения параметров должны копироваться с

одного компьютера на другой. Для копирования параметров процедуры и результата выполнения через сеть выполняется их сериализация.

- В отличие от локального вызова удаленный вызов процедур обязательно использует нижележащую систему связи, однако это не должно быть явно видно ни в определении процедур, ни в самих процедурах.
- Выполнение вызывающей программы и вызываемой локальной процедуры в одной машине реализуется в рамках единого процесса. Но в реализации RPC участвуют как минимум два процесса — по одному в каждой машине. В случае, если один из них аварийно завершится, могут возникнуть следующие ситуации: при аварии вызывающей процедуры удаленно вызванные процедуры станут «осиротевшими», а при аварийном завершении удаленных процедур станут «обездоленными родителями» вызывающие процедуры, которые будут безрезультатно ожидать ответа от удаленных процедур.
- Существует ряд проблем, связанных с неоднородностью языков программирования и операционных сред: структуры данных и структуры вызова процедур, поддерживаемые в каком-либо одном языке программирования, не поддерживаются точно так же во всех других языках.

Эти и некоторые другие проблемы решает широко распространенная технология RPC, лежащая в основе многих распределенных операционных систем.

XDR (англ. *External Data Representation*) — стандарт передачи данных в Интернете, используемая в различных RFC для описания типов. XDR используют следующие программы:

- Sun RPC
- NetCDF
- Язык программирования R
- SpiderMonkey
- Ganglia

Типы данных в XDR

- boolean
- int (32-ное целое число)
- hyper (32-ное целое число)
- float
- double
- enumeration
- structure
- string
- массивы фиксированной длины
- массивы переменной длины
- неформатированные ("сырые") данные

15.Понятие “socket”. Службы, вызовы, принципы работы.

Со́кеты (англ. *socket* углубление, гнездо, разъём) — это название программного интерфейса для обеспечения информационного обмена между процессами. Процессы при таком обмене могут исполняться как на одной ЭВМ, так и на различных ЭВМ, связанных между собой сетью. Сокет — абстрактный объект, представляющий конечную точку соединения.

Следует различать **клиентские** и **серверные сокеты**. Клиентские сокеты грубо можно сравнить с оконечными апаратами телефонной сети, а серверные — с коммутаторами. Клиентское приложение (например, браузер) использует только клиентские сокеты, а серверное (например, веб-сервер, которому браузер посылает запросы) — как клиентские, так и серверные сокеты.

Интерфейс сокетов впервые появился в BSD Unix. Программный интерфейс сокетов описан в стандарте POSIX.1 и в той или иной мере поддерживается *всеми* современными операционными системами.

Сокет, на сленге современных системных администраторов, означает комбинацию IP-адреса и номера порта, например 10.10.10.10:80.

Принципы сокетов

Каждый процесс может создать *слушающий* сокет (серверный сокет) и *привязать* его к какому-нибудь порту компьютера (тем не менее, в UNIX непривилегированные процессы не могут использовать порты меньше 1024). Слушающий процесс обычно находится в цикле ожидания, то есть просыпается при появлении нового соединения. При этом сохраняется возможность просто проверить наличие соединений на данный момент, установить тайм-аут для операции и так далее.

Каждый сокет имеет свой адрес. ОС семейства UNIX могут поддерживать много типов адресов, но обязательными являются INET-адрес и UNIX-адрес. Если привязать сокет к UNIX-адресу, то просто будет создан специальный файл (*файл сокета*) по заданному пути, через который смогут сообщаться любые локальные процессы путём простого чтения/записи из него. Сóкеты типа INET доступны из сети и требуют выделения номера порта.

Обычно клиент явно *подсоединяется* к слушателю, после чего любое чтение или запись через его файловый дескриптор будут на самом деле передавать данные между ним и сервером.

16. Маршрутизация: маршрутизация первого уровня.

17. Маршрутизация: маршрутизация второго уровня.

18. Маршрутизация: маршрутизация третьего уровня.

19. Маршрутизация: основные понятия, уровни маршрутизации.

Большинство производителей коммутаторов рзличают следующие классы устройств:

1. Неуправляемые коммутаторы 2-го уровня (иногда их по ошибке называют коммутаторами 1-го уровня). Применение - небольшая сеть из 3-10 компьютеров.

2. Интеллектуальные (конфигурируемые, smart) коммутаторы 2-го уровня. Применение - сети малых предприятий.

3. Управляемые L2 (иногда их по ошибке называют коммутаторами 2-го уровня). Применение - сети средних и крупных предприятий и провайдеров для подключения юзеров.

4. Управляемые L2+ (или L2-4). Чаще их используют провайдеры (ACL, снижение скорости), иногда предприятия, где есть IP-телефония или критичекие к задержкам приложения.

5. Управляемые 3-го уровня (базового 3-го уровня). Применение - ядро сети небольшого предприятия.

6. Многоуровневые (L3+, L3-7). Применение - ядро сети крупного предприятия, провайдеры

Это уровни согласно OSI

1-й уровень - физический

Понятия: ток, напряжение, сопротивление.

Репитер - устройство первого уровня.

2-й уровень - канальный

Понятия: кадр, он же фрейм.

Свич, мост - устройства второго уровня.

3-й уровень - сетевой ()

Понятия: Пакет.

Роутер - устройства третьего уровня.

Часто в свичах (которые по определению второго уровня) имеются функции устройств третьего уровня (типа поддержка spanning tree, фильтрация пакетов и т.д.). В этом случае, АФАЙК, они называются свичами трельтего уровня.

20. Протоколы маршрутизации: RIP, OSPF, BGP, EGP. Сравнительные характеристики.

Протокол маршрутизации — это сетевой протокол, используемый маршрутизаторами для определения возможных маршрутов следования данных в составной компьютерной сети. Применение протокола маршрутизации позволяет избежать ручного ввода всех допустимых маршрутов, что, в свою очередь, снижает количество ошибок, обеспечивает согласованность действий всех маршрутизаторов в сети и облегчает труд администраторов.

Протоколы маршрутизации делятся на два вида, зависящие от типов алгоритмов, на которых они основаны:

- Дистанционно-векторные протоколы, основаны на Distance Vector Algorithm (DVA);
- Протоколы состояния каналов связи, основаны на Link State Algorithm (LSA).

Дистанционно-векторные протоколы:

- RIP - Routing Information Protocol
- IGRP - Interior Gateway Routing Protocol (лицензированный протокол Cisco Systems)
- BGP - Border GateWay Protocol
- EIGRP - Enhanced Interior Gateway Routing Protocol (лицензированный протокол Cisco Systems)

Протоколы состояния каналов связи:

- IS-IS - Intermediate System to Intermediate System (стек OSI)
- OSPF - Open Shortest Path First
- NLSP - NetWare Link-Services Protocol (стек Novell)

и на два вида в зависимости от сферы применения: для междоменной и внутридоменной маршрутизации:

Междоменная маршрутизация:

- EGP
- BGP
- IDRP
- IS-IS level 2

Внутридоменная маршрутизация:

- RIP
- IS-IS level 1
- OSPF
- IGRP
- EIGRP

21. Протокол маршрутизации RIP.

Алгоритм маршрутизации RIP (алгоритм Беллмана — Форда) был впервые разработан в 1969 г., как основной для сети ARPANET.

Прототип протокола RIP — Gateway Information Protocol, часть пакета PARC Universal Packet.

Версия RIP, которая поддерживает протокол интернета была включена в пакет BSD операционной системы Unix под названием routed (route daemon), а также многими производителями, реализовавшими свою версию этого протокола. В итоге протокол был унифицирован в документе RFC 1058.

В 1994 г. был разработан протокол RIP2 (RFC 2453), который является расширением протокола RIP, обеспечивающим передачу дополнительной маршрутной информации в сообщениях RIP и повышающим уровень безопасности.

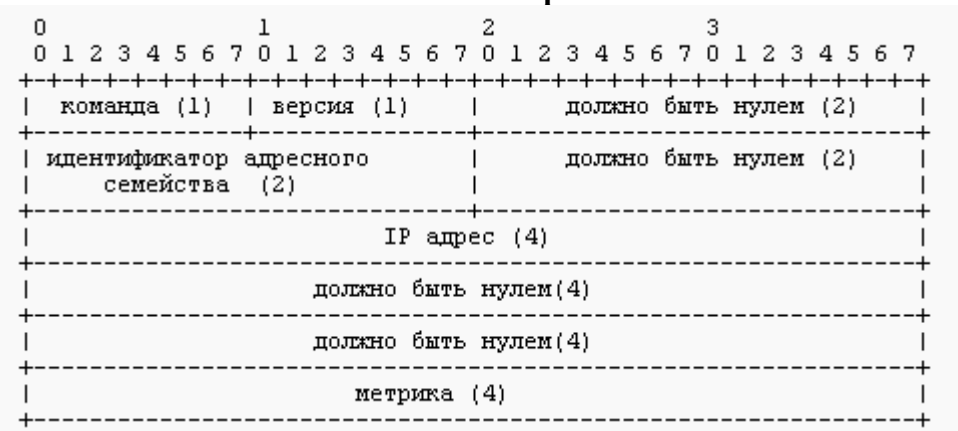
Для работы в среде IPv6 была разработана версия RIPng.

Техническая информация

RIP — так называемый протокол вектор-расстояния, который оперирует хопами в качестве метрики маршрутизации. Максимальное количество хопов, разрешенное в RIP — 15 (метрика 16 означает "бесконечно большую метрику"). Каждый RIP-маршрутизатор по умолчанию вещает в сеть свою полную таблицу маршрутизации раз в 30 секунд, генерируя довольно много трафика на низкоскоростных линиях связи. RIP работает на сетевом уровне стека TCP/IP, используя UDP порт 520.

В современных сетевых средах RIP — не самое лучшее решение для выбора в качестве протокола маршрутизации, так как его возможности уступают более современным протоколам, таким как EIGRP, OSPF. Ограничение на 15 хопов не дает применять его в больших сетях. Преимущество этого протокола —- простота конфигурирования.

Формат RIP пакета



22.Протокол маршрутизации OSPF

OSPF (англ. Open Shortest Path First) — протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала (link-state technology) и использующий для нахождения кратчайшего пути Алгоритм Дейкстры (Dijkstra's algorithm).

Протокол OSPF был разработан IETF в 1988 году. Последняя версия протокола представлена в RFC 2328. Протокол OSPF представляет собой протокол внутреннего шлюза (Interior Gateway Protocol — IGP). Протокол OSPF распространяет информацию о доступных маршрутах между маршрутизаторами одной автономной системы.

OSPF предлагает решение следующих задач:

- Увеличение скорости сходимости;
- Поддержка сетевых масок переменной длины (VLSM);
- Достижимость сети;
- Использование пропускной способности;
- Метод выбора пути.

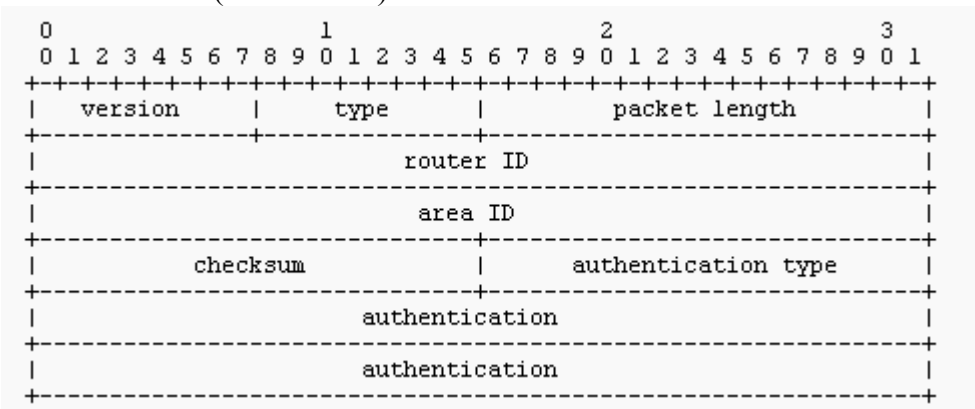
Описание работы протокола

- Маршрутизаторы обмениваются hello-пакетами через все интерфейсы на которых активирован OSPF. Маршрутизаторы разделяющие общий канал передачи данных становятся соседями, когда они приходят к договоренности об определенных параметрах указанных в их hello-пакетах.
- На следующем этапе работы протокола маршрутизаторы будут пытаться перейти в состояние соседства со своими соседями. Переход в состояние соседства определяется типом маршрутизаторов обменивающихся hello-пакетами и типом сети по которой передаются hello-пакеты. OSPF определяет несколько типов сетей и несколько типов маршрутизаторов. Пара маршрутизаторов, находящихся в состоянии соседства синхронизирует между собой базу данных состояния каналов.

- Каждый маршрутизатор посылает объявление о состоянии канала маршрутизаторам с которыми он находится в состоянии соседства.
- Каждый маршрутизатор получивший объявление от соседа записывает информацию передаваемую в нем в базу данных состояния каналов маршрутизатора и рассылает копию объявления всем другим своим соседям.
- Рассылая объявления через зону, все маршрутизаторы строят идентичную базу данных состояния каналов маршрутизатора.
- Когда база данных построена, каждый маршрутизатор использует алгоритм кратчайший путь первым для вычисления графа без петель, который будет описывать кратчайший путь к каждому известному пункту назначения с собой в качестве корня. Этот граф это дерево кратчайшего пути.
- Каждый маршрутизатор строит таблицу маршрутизации из своего дерева кратчайшего пути.

Типы сетей, поддерживаемые протоколом OSPF

- Широковещательные сети со множественным доступом (Ethernet, Token Ring)
- Точка-точка (T1, E1, коммутируемый доступ)
- Нешироковещательные сети со множественным доступом (NBMA) (Frame relay)
- Виртуальные каналы (virtual links)



OSPF-пакет инкапсулируется непосредственно в поле данных IP-пакета. Значение поля «протокол верхнего уровня» в заголовке IP-дейтаграммы для OSPF равно 89.
version — номер версии протокола OSPF. Текущая версия OSPF для сетей IPv4 — 2.
type — тип OSPF-пакета. В RFC 2328 описано 5 типов пакетов.
packet length — длина пакета, включая заголовок.
router ID — идентификатор маршрутизатора — уникальное 32-хбитное число, идентифицирующее
маршрутизатор в пределах автономной системы.
area ID — 32-хбитный идентификатор зоны.
checksum — поле контрольной суммы. Подсчитывается для всего пакета, включая заголовок.
authentication type — тип используемой схемы аутентификации. Возможные значения:
0 — аутентификация не используется
1 — аутентификация открытым текстом
2 — MD5-аутентификация
authentication — поле данных аутентификации.

23.Протокол маршрутизации BGP

BGP (англ. Border Gateway Protocol, протокол граничного шлюза) — основной протокол динамической маршрутизации в Интернете.

BGP, в отличие от других протоколов динамической маршрутизации, предназначен для обмена информацией о маршрутах не между отдельными маршрутизаторами, а между целыми автономными системами, и поэтому, помимо информации о маршрутах в сети, переносит также информацию о маршрутах на автономные системы. BGP не использует технические метрики, а осуществляет выбор наилучшего маршрута исходя из правил, принятых в сети.

BGP поддерживает бесклассовую адресацию и использует суммирование маршрутов для уменьшения таблиц маршрутизации. С 1994 года действует четвёртая версия протокола, все предыдущие версии являются устаревшими.

BGP является протоколом сетевого уровня и функционирует поверх протокола транспортного уровня TCP (порт 179).

BGP, наряду с DNS, является одним из главных механизмов, обеспечивающих функционирование Internet.

24.Протокол маршрутизации EGP

EGP (сокр. от англ. Exterior Gateway Protocol, протокол внешнего шлюза) — устаревший протокол обмена информации между маршрутизаторами нескольких автономных систем. Разработан в 82-84 годах. В последствии был заменён на BGP.

Типы сообщений

Request — Запрос на захват соседей и/или установки настроек опроса
Confirm — Подтверждение запроса request
Refuse — Отказ в захвате соседей
Cease — Запрос на перезахват соседей
Cease-ack — подтверждение перезахвата соседей
Hello — проверка доступности соседей
I-H-U — ответ на запрос о доступности (англ. I hear you, я тебя слышу)
Poll — запрос на обновление информации о доступности сети
Update — обновление информации о доступности сети
Error — ошибка

Формат заголовка сообщениябайт значение

1	версия (англ. version)
2	тип сообщения (англ. type)
3	код сообщения (англ. code)
4	статус (англ. status
5-6	контрольная сумма (2 байта)
7-8	Номер автононой системы
9-10	Порядковый номер сообщения

25. Сети X.25.

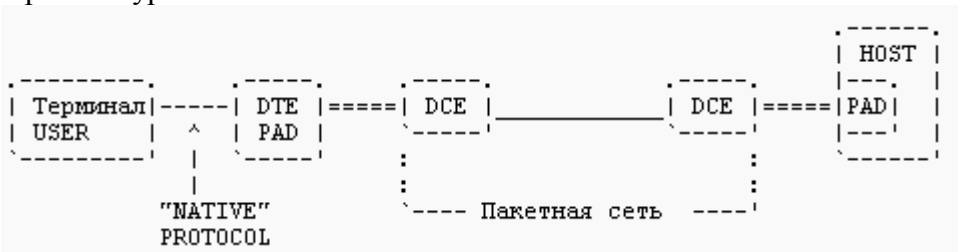
X.25 — семейство протоколов канального уровня сетевой модели OSI. Предназначался для организации WAN на основе телефонных сетей с линиями с достаточно высокой частотой ошибок, поэтому содержит развитые механизмы коррекции ошибок. Ориентирован на работу с установлением соединений, Исторически является предшественником протокола Frame Relay.

X.25 обеспечивает множество независимых виртуальных каналов (Permanent Virtual Circuits, PVC и Switched Virtual Circuits, SVC) в одной линии связи, идентифицируемых в X.25-сети по идентификаторам подключения к соединению (идентификаторы логического

канала (Logical Channel Identifyer, LCI) или номера логического канала (Logical Channel Number, LCN).

Благодаря надёжности протокола и его работе поверх телефонных сетей общего пользования X.25 широко использовался как в корпоративных сетях, так и во всемирных специализированных сетях предоставления услуг, таких как SWIFT (банковская платёжная система) и SITA (фр. Société Internationale de Télécommunications Aéronautiques — система информационного обслуживания воздушного транспорта), однако в настоящее время X.25 вытесняется другими технологиями канального уровня (Frame Relay, ISDN, ATM) и протоколом IP, оставаясь, однако, достаточно распространённым в странах и территориях с неразвитой телекоммуникационной инфраструктурой.

Архитектура



Режимы и типы пакетов X.25

Режим установления соединения (Call setup mode) используется при установлении соединения SVC между DTE-устройствами. В этом режиме на уровне PLP используется схема адресации X.121 для установления виртуального соединения. Режим установления соединения работает на уровне виртуальных каналов, то есть в пределах одного физического DTE-устройства одни SVC могут быть в состоянии установления соединения, а другие — в режиме передачи данных или разрыва соединения. Режим установления соединения используется только в случае установления SVC, но не PVC.

Режим передачи данных (Data transfer mode) используется при передаче данных по виртуальному каналу. При этом X.25 PLP ответственнен за сегментацию данных в пакеты и сборку пакетов, управление передачей данных и коррекцию ошибок. Режим передачи данных работает на уровне виртуальных каналов и используется в случае как SVC, так и PVC.

Режим ожидания (Idle mode) характеризуется отсутствием передачи данных при установленном виртуальном канале. Работает на уровне виртуальных каналов и используется только в случае установления SVC, но не PVC.

Режим разрыва соединения (Call clearing mode) используется при разрыве соединения SVC между DTE-устройствами. Работает на уровне виртуальных каналов и используется только в случае разрыва SVC, но не PVC.

Режим перезапуска (Restarting mode) используется для переустановления соединений между DTE-устройством и локально работающих с ним DCE-устройствами. В отличие от других режимов, выполняется пределах одного физического DTE-устройства, что сопровождается разрывом всех виртуальных каналов, установленных с этим DTE.

26.Сети Frame Relay.

Frame relay (англ. «ретрансляция кадров», FR) — протокол канального уровня сетевой модели OSI. Служба коммутации пакетов Frame Relay в настоящее время широко распространена во всём мире. Максимальная скорость, допускаемая протоколом FR — 34.368 мегабит/сек (каналы E3).

Frame Relay был создан в начале 1990-х в качестве замены протоколу X.25 для быстрых надёжных каналов связи, технология FR архитектурно основывалась на X.25 и во многом сходна с этим протоколом, однако в отличие от X.25, рассчитанного на линии с достаточно высокой частотой ошибок, FR изначально ориентировался на физические линии с низкой частотой ошибок, и поэтому большая часть механизмов коррекции ошибок X.25 в состав

стандарта FR не вошла. В разработке спецификации принимали участие многие организации; многочисленные поставщики поддерживают каждую из существующих реализаций, производя соответствующее аппаратное и программное обеспечение.

Frame relay обеспечивает множество независимых виртуальных каналов (Permanent Virtual Circuits, PVC) в одной линии связи, идентифицируемых в FR-сети по идентификаторам подключения к соединению (Data Link Connection Identifier, DLCI), но не имеет средств коррекции и восстановления. Вместо средств управления потоком включает функции извещения о перегрузках в сети. Возможно назначение минимальной гарантированной скорости (CIR) для каждого виртуального канала.

В основном применяется при построении территориально распределённых корпоративных сетей, а также в составе решений, связанных с обеспечением гарантированной пропускной способности канала передачи данных (VoIP, видеоконференции и т. п.).

Формат кадра Флаг (1 Byte) Адрес (2-4 Byte) Данные (переменный размер) FCS (2 Byte) Флаг (1 Byte)

- Каждый кадр начинается и замыкается «флагом» — последовательностью «01111110». Для предотвращения случайной имитации последовательности «флаг» внутри кадра при его передаче проверяется всё его содержание между двумя флагами и после каждой последовательности, состоящей из пяти идущих подряд бит «1», вставляется бит «0». Эта процедура (bit stuffing) обязательна при формировании любого кадра FR, при приёме эти биты «0» отбрасываются.
- FCS (Frame Check Sequence) — проверочная последовательность кадра служит для обнаружения ошибок и формируется аналогично циклическому коду HDLC.
- Поле данных имеет минимальную длину в 1 октет, максимальную по стандарту Frame Relay Forum — 1600 октетов, однако в реализациях некоторых производителей FR-оборудования допускается превышение максимального размера (до 4096 октетов).
- Поле Адрес кадра Frame Relay, кроме собственно адресной информации, содержит также и дополнительные поля управления потоком данных и уведомлений о перегрузке канала

27.Сети ATM.

ATM (англ. Asynchronous Transfer Mode — асинхронный способ передачи данных) — сетевая технология, основанная на передаче данных в виде ячеек (cell) фиксированного размера (53 байта), из которых 5 байтов используется под заголовок. Ячейки данных, используемые в ATM, меньше в сравнении с элементами данных, которые используются в других технологиях. Небольшой, постоянный размер ячейки, используемый в ATM, позволяет:

- передавать данные по одним и тем же физическим каналам, причём как при низких, так и при высоких скоростях;
- работать с постоянными и переменными потоками данных;
- интегрировать любые виды информации: тексты, речь, изображения, видеофильмы;
- поддерживать соединения типа точка-точка, точка-многоточка, многоточка-многоточка.

Для передачи данных от отправителя к получателю в сети ATM создаются виртуальные каналы, VC (англ. Virtual Circuit), которые бывают двух видов:

- постоянный виртуальный канал, PVC (Permanent Virtual Circuit), который создаётся между двумя точками и существует в течение длительного времени, даже в отсутствие данных для передачи;

- коммутируемый виртуальный канал, SVC (Switched Virtual Circuit), который создаётся между двумя точками непосредственно перед передачей данных и разрывается после окончания сеанса связи.

28. Сети хранения данных – основные понятия, определения и термины.

Сеть хранения данных (СХД) (англ. Storage Area Network) — представляет собой архитектурное решение для подключения внешних устройств хранения данных, таких как дисковые массивы, ленточные библиотеки, оптические накопители к серверам, таким образом, чтобы операционная система распознала подключённые ресурсы, как локальные. Несмотря на то, что стоимость и сложность таких систем постоянно падают, по состоянию на 2007 год сети хранения данных остаются редкостью за пределами больших предприятий.

В отличие от SAN, сетевые хранилища данных (NAS) используют сетевые протоколы для доступа к файлам (такие как NFS или SMB/CIFS); при использовании этих протоколов понятно, что хранилище является удалённым и компъютер запрашивает файл вместо того, чтобы запрашивать блок данных с диска.

Типы сетей

Большинство сетей хранения данных использует протокол SCSI для связи между серверами и устройствами хранения данных на уровне шинной топологии. Так как протокол SCSI не предназначен для формирования сетевых пакетов, в сетях хранения данных используются низкоуровневые протоколы:

- Fibre Channel Protocol (FCP), транспорт SCSI через Fibre Channel. Наиболее используемый на данный момент протокол. Существует в вариантах 1 Gbit/s, 2 Gbit/s, 4 Gbit/s, 8 Gbit/s, 10 Gbit/s.
- iSCSI, транспорт SCSI через TCP/IP.
- HyperSCSI, транспорт SCSI через Ethernet.
- FICON транспорт через Fibre Channel (используется мейнфреймами).
- ATA over Ethernet, транспорт ATA через Ethernet.
- SCSI и/или TCP/IP транспорт через InfiniBand (IB).

Совместное использование устройств хранения

Движущей силой для развития сетей хранения данных стал взрывной рост объема деловой информации (такой как электронная почта, базы данных и высоконагруженные файловые сервера), требующей высокоскоростного доступа к дисковым устройствам на блочном уровне . Ранее на предприятии возникали "острова" высокопроизводительных дисковых массивов SCSI. Каждый такой массив был выделен для конкретного приложения и виден ему как некоторое количество "виртуальных жестких дисков" (LUN'ов).

Сеть хранения данных позволяет объединить эти "острова" средствами высокоскоростной сети.

Сети хранения помогают повысить эффективность использования ресурсов систем хранения, поскольку дают возможность выделить любой ресурс любому узлу сети.

Преимущества

Совместное использование систем хранения как правило упрощает администрирование и добавляет изрядную гибкость, поскольку кабели и дисковые массивы не нужно физически транспортировать и перекоммутировать от одного сервера к другому.

Другим приемуществом является возможность загружать сервера прямо из сети хранения. При такой конфигурации можно быстро и легко заменить сбойный сервер, переконфигурировав SAN таким образом, что сервер-замена, будет загружаться с LUN'a сбойного сервера. Эта процедура может занять, например, полчаса. Идея относительно новая, но уже используется в новейших датацентрах.

Также сети хранения помогают более эффективно восстанавливать работоспособность после сбоя. В SAN может входить удаленный участок со вторичным устройством хранения. В таком случае можно использовать репликацию - реализованную на уровне контроллеров массивов, либо при помощи специальных аппаратных устройств. Поскольку каналы WAN на основе протокола IP встречаются часто, были разработаны протоколы Fibre Channel over IP (FCIP) и iSCSI с целью расширить единую SAN средствами сетей на основе протокола IP. Спрос на такие решения значительно возрос после событий 11 сентября 2001 года в США.

29. Дисковые устройства хранения данных.

30. Ленточные устройства хранения данных.

31. Оптические и магнито-оптические устройства хранения данных.

32. Дисковые массивы: JBOD, RAID.

JBOD (от англ. Just a bundle of disks, просто куча дисков) — рейд-массив дисков, в которых дисковое пространство распределено по жёстким дискам последовательно.

Пример распределения файлов по JBOD-массиву (разные файлы выделены разными цветами)

Характеристики JBOD массива

- Ёмкость массива равна сумме ёмкостей составляющих дисков
- Вероятность отказа приблизительно равна сумме вероятностей отказа каждого диска в массиве (избыточность не предусмотрена)
- Скорость чтения и записи зависит от области данных; она не выше, чем у самого быстрого диска в массиве и не ниже чем у самого медленного
- Нагрузка на процессор при работе минимальная (сравнимая с нагрузкой при работе с единичным диском)

Особенности JBOD массива

- Отказ одного диска позволяет восстановить файлы на остальных дисках (если их начало/конец не принадлежат повреждённому диску)
- В ряде случаев возможно обеспечение высокой скорости работы нескольких приложений (при условии, что приложения работают с областями данных на разных дисках)
- Массив может состоять из дисков различной ёмкости и скорости
- Массив легко расширяется дополнительными дисками по мере надобности

В операционной системе Windows JBOD-массив называется spanned disk (возможно создание только на динамических дисках).

RAID (англ. redundant array of independent/inexpensive disks) — дисковый массив независимых дисков. Служит для повышения надёжности хранения данных и/или для повышения скорости чтения/записи информации (RAID 0).

Пример простейшего PCI IDE RAID контроллера. В этом устройстве все функции RAID выполняет программный драйвер ОС. Поддерживаются только режимы RAID 0 и RAID 1

Аббревиатура RAID изначально расшифровывалась как «Redundant Arrays of Inexpensive Disks» («избыточный (резервный) массив недорогих дисков», так как они были

гораздо дешевле RAM). Именно так был представлен RAID его создателями Петтерсоном (David A. Patterson), Гибсоном (Garth A. Gibson) и Катцом (Randy H. Katz) в 1987 году. Со временем RAID стали расшифровывать как «Redundant Array of Independent Disks» («избыточный (резервный) массив независимых дисков»), потому как для массивов приходилось использовать и дорогое оборудование (под недорогими дисками подразумевались диски для ПЭВМ).

33. Дисковые массивы с RAID: уровни RAID, принципы организации по уровням.

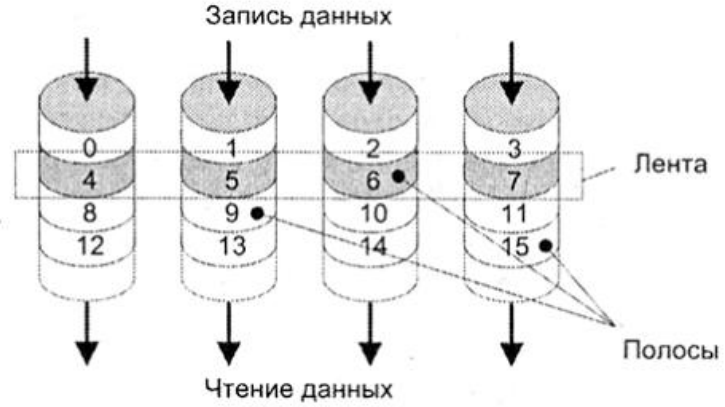


Рис.33 RAID уровня 0

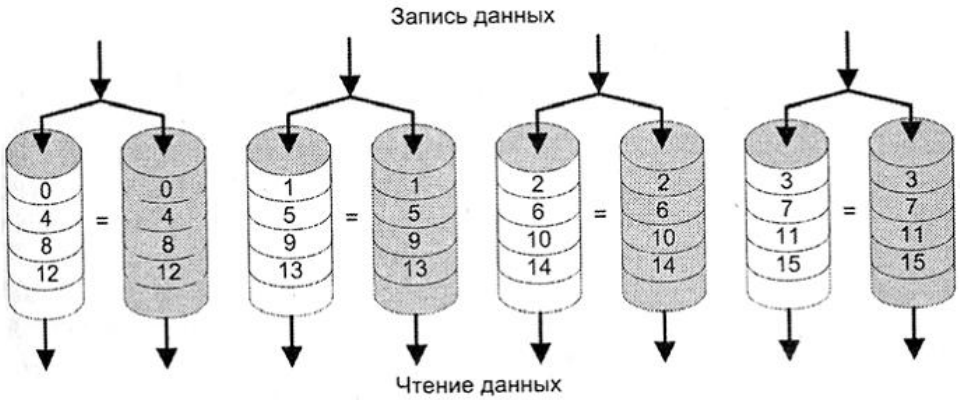


Рис.34 RAID уровня 1

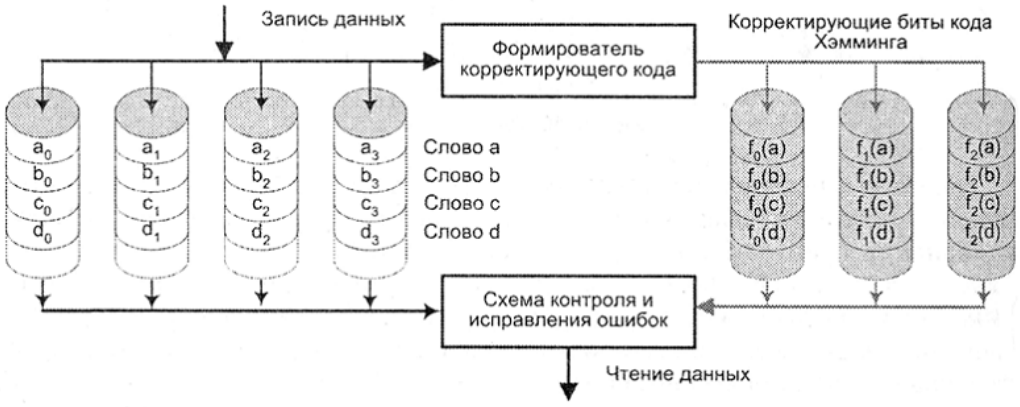


Рис.35 RAID уровня 2

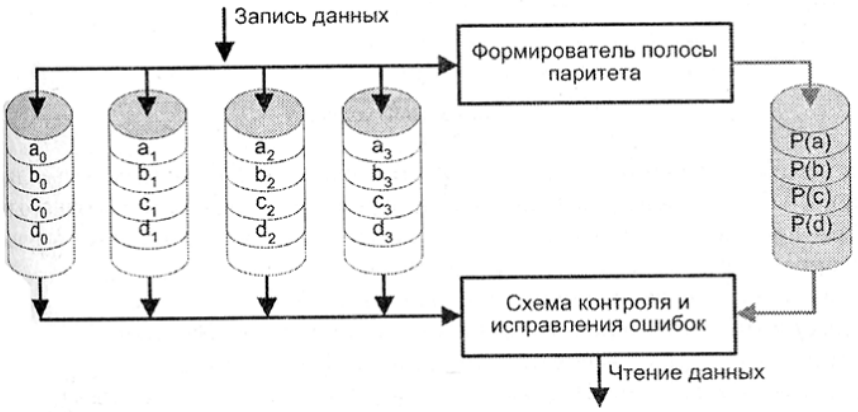


Рис.36 RAID уровня 3

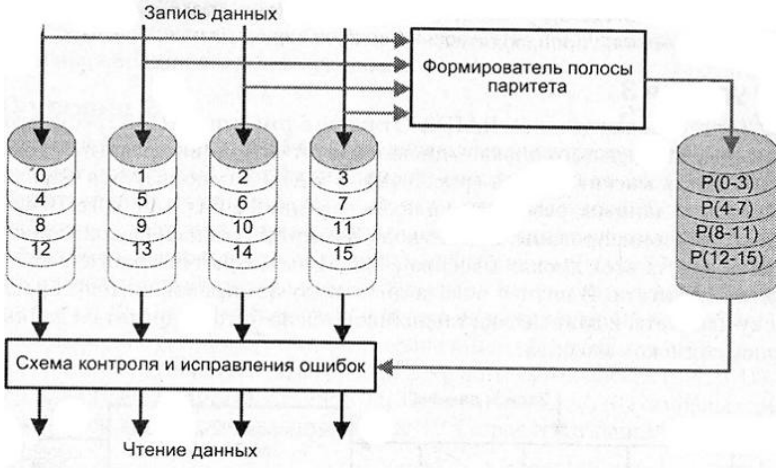


Рис.37 RAID уровня 4

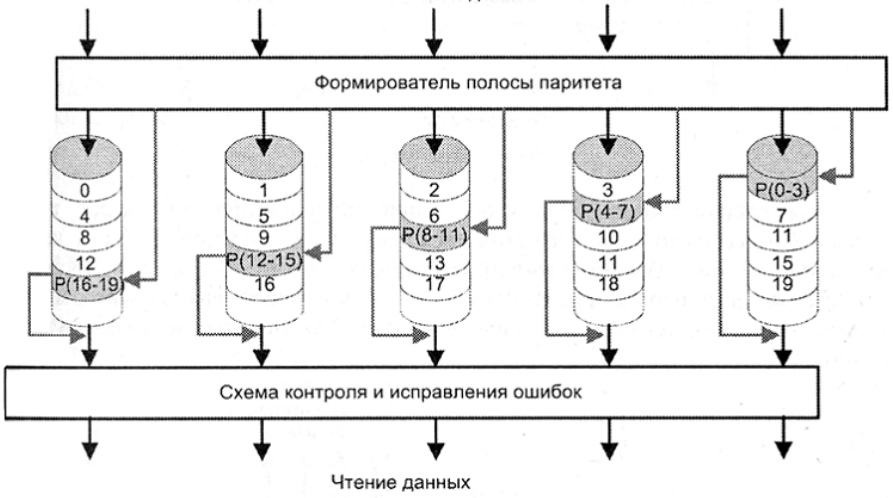
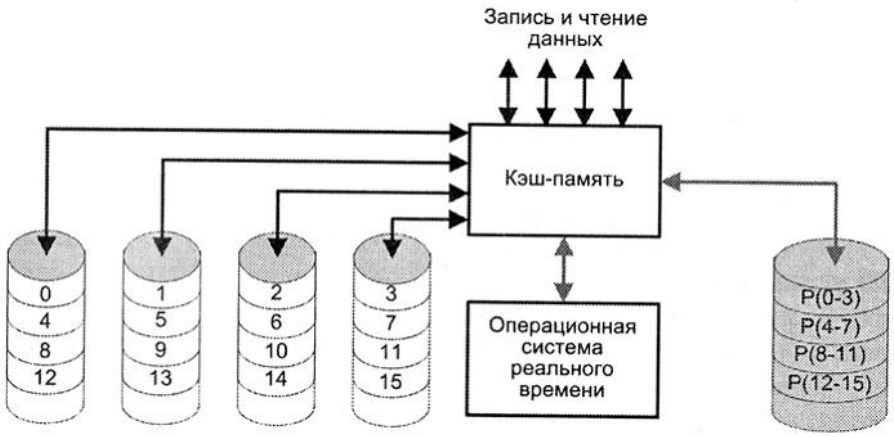


Рис.38 RAID уровня 5



уровень 7

34. Сравнительные характеристики протоколов в системах хранения данных. Сравнительные характеристики.

Наиболее перспективными технологиями построения корпоративных хранилищ данных сегодня признаны сетевые устройства хранения SAN (Storage Area Network), устройства прямого подключения к серверам DAS (Direct Attached Storage) и устройства подключаемые через интернет NAS (Network Attached Storage).

Сравнительные характеристики систем хранения данных

Характеристика	**NAS**	**SAS (DAS)**	**SAN**
Протоколы передачи данных	CIFS, HTTP, NFS, FTP	SCSI, SSA	SCSI
Скорость передачи	не менее 100 МБ/с на один порт	несколько сот МБ/с	до 1 Гб/с на один порт
Сетевые протоколы	TCP/IP через Ethernet, FDDI, ATM, Gigabit Ethernet	SCSI-интерфейс сервера, сетевой протокол неприемлем	Fibre Channel, Gigabit Ethernet
Масштабирование	Качественное, но снижает пропускную способность сети	Ограничено количеством подключаемых устройств и производительностью единственного сервера	Самое эффективное
Миграция данных	Используются способы резервирования/ восстановления	Снижает производительность сервера	Обеспечивается построение систем хранения высокой готовности с возможностью дублирования в реальном времени

35. Протокол Parallel SCSI.

Ключевым отличием SAS от SCSI является возможность подключения SAS-накопителей одновременно к двум различным портам, каждый из которых представляет различные доме-

ны SAS. Можете себе представить, насколько значительным образом это отражается на надежности хранения данных и отказоустойчивости системы. К тому же, "коммутаторная" природа архитектуры SAS позволяет в теории подключать "покаскадно" тысячи накопителей (до 16384 приводов без снижения производительности!), что делает масштабируемость таких систем теоретически неограниченной. Основные отличия технологий SCSI и SAS приведены в таблице ниже.

	Parallel SCSI	SAS
Архитектура	Параллельная, все устройства подключаются к распределенной шине	Последовательная, "точка-точка", дискретные сигнальные пути
Производительность	320 МБ/с (Ultra320 SCSI); производительность падает по мере добавления приводов	3.0 Гб/с, в планах до 12.0 Гб/с; производительность не зависит от числа дисков
Масштабируемость	15 накопителей	Более 16000 накопителей
Совместимость	Не совместимо с другими интерфейсами	Совместимо с Serial ATA (SATA)
Мах. длина кабеля	12 м, суммируется длина всех кабелей на шине	8 м на отдельный порт; в сумме - до сотен метров
Форм-фактор кабеля	Множество проводников	Компактные разъемы и шины
"Горячее" подключение	Нет	Есть
Идентификация приводов	Устанавливается вручную, необходима уверенность в отсутствии конфликтов ID на шине	Набор мировых уникальных ID от производителя
Терминация	Устанавливается вручную, необходима уверенность в правильной инсталляции и работе терминаторов	Дискретные сигнальные пути по умолчанию; не требует вмешательства пользователя

36.Протокол iSCSI.

iSCSI (Internet Small Computer System Interface) — это протокол, который базируется на TCP/IP и разработан для установления взаимодействия и управления системами хранения данных, серверами и клиентами».

Определение SNIA — IP Storage Forum: http://www.snia.org/

iSCSI описывает:

- Транспортный протокол для SCSI, который работает поверх TCP
- Новый механизм инкапсуляции SCSI команд в IP сети
- Протокол для новой генерации систем хранения данных, которые будут использовать «родной» TCP/IP

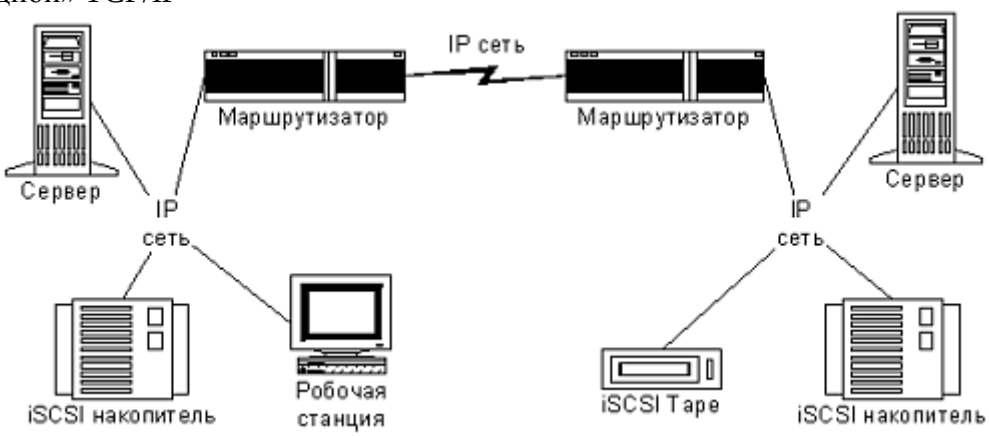


Рисунок 1. IP сеть с использованием iSCSI устройств

В примере, изображенном на рисунке 1, каждый сервер, рабочая станция и накопитель поддерживают Ethernet интерфейс и стек протокола iSCSI. Для организации сетевых соединений используются IP маршрутизаторы и Ethernet коммутаторы.

С внедрением SAN мы получили возможность использовать SCSI протокол в сетевых инфраструктурах, обеспечивая высокоскоростную передачу данных на уровне блоков между множественными элементами сети хранения данных.

Internet Small Computer System Interface тоже обеспечивает блочный доступ к данным, но не самостоятельно, а поверх сетей TCP/IP.

Архитектура обычного SCSI базируется на «клиент»/«серверной» модели. «Клиент», например сервер, или рабочая станция, инициирует запросы на считывание или запись данных с исполнителя — «сервера», например системы хранения данных. Команды, которые выдает «клиент» и обрабатывает «сервер» помещаются в Command Descriptor Block (CDB). «Сервер» выполняет команду, а окончание ее выполнения обозначается специальным сигналом. Инкапсуляция и надежная доставка CDB транзакций между инициаторами и исполнителями через TCP/IP сеть и есть главная задача iSCSI, причем ее приходится осуществлять в нетрадиционной для SCSI, потенциально ненадежной среде IP сетей.

Перед вами модель уровней протокола iSCSI, которая дает возможность понять порядок инкапсуляции SCSI команд для передачи их через физический носитель.

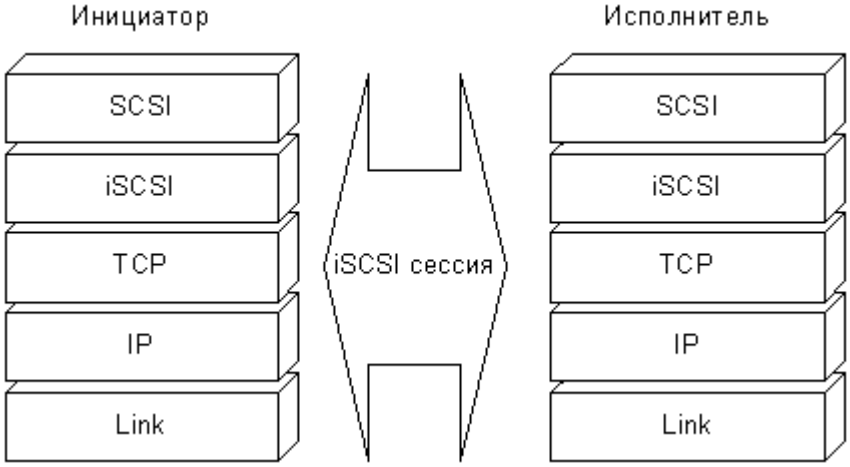


Рисунок 2. Модель нижних уровней протокола iSCSI

iSCSI протокол осуществляет контроль передачи блоков данных и обеспечивает подтверждение достоверности завершения операции ввода/вывода. Что, в свою очередь, обеспечивается через одно или несколько TCP соединений.

iSCSI имеет четыре составляющие:

- Управление именами и адресами (iSCSI Address and Naming Conventions).
- Управление сеансом (iSCSI Session Management).
- Обработка ошибок (iSCSI Error Handling).
- Безопасность (iSCSI Security).

37.Протокол Parallel ATA.

Для подключения жёстких дисков с интерфейсом PATA обычно используется 40-проводный кабель (именуемый также шлейфом). Каждый шлейф обычно имеет два или три разъёма, один из которых подключается к разъёму контроллера на материнской плате (в более старых компьютерах этот контроллер размещался на отдельной плате расширения), а один или два других подключаются к дискам. В один момент времени шлейф P-ATA переда-

ёт 16 бит данных. Иногда встречаются шлейфы IDE, позволяющие подключение трёх дисков к одному IDE каналу, но в этом случае один из дисков работает в режиме read-only.

Разводка Parallel ATA

Контакт	Назначение	Контакт	Назначение
1	Reset	2	Ground
3	Data 7	4	Data 8
5	Data 6	6	Data 9
7	Data 5	8	Data 10
9	Data 4	10	Data 11
11	Data 3	12	Data 12
13	Data 2	14	Data 13
15	Data 1	16	Data 14
17	Data 0	18	Data 15
19	Ground	20	Key
21	DDRQ	22	Ground
23	I/O Write	24	Ground
25	I/O Read	26	Ground
27	IOC HRDY	28	Cable Select
29	DDACK	30	Ground
31	IRQ	32	No Connect
33	Addr 1	34	GPIO_DMA66_Detect
35	Addr 0	36	Addr 2
37	Chip Select 1P	38	Chip Select 3P
39	Activity	40	Ground

Долгое время шлейф ATA содержал 40 проводников, но с введением режима Ultra DMA/66 (UDMA4) появилась его 80-проводная версия. Все дополнительные проводники — это проводники заземления, чередующиеся с информационными проводниками. Такое чередование проводников уменьшает ёмкостную связь между ними, тем самым сокращая взаимные наводки. Ёмкостная связь являются проблемой при высоких скоростях передачи, поэтому данное нововведение было необходимо для обеспечения нормальной работы установленной спецификацией UDMA4 скорости передачи 66 МБ/с (мегабайт в секунду). Более быстрые режимы UDMA5 и UDMA6 также требуют 80-проводного кабеля.

Хотя число проводников удвоилось, число контактов осталось прежним, как и внешний вид разъёмов. Внутренняя же разводка, конечно, другая. Разъёмы для 80-проводного кабеля должны присоединять большое число проводников заземления к небольшому числу контактов заземления, в то время, как в 40-проводном кабеле проводники присоединяются каждый к своему контакту. У 80-проводных кабелей разъёмы обычно имеют различную расцветку (синий, серый и чёрный), в отличие от 40-проводных, где обычно все разъёмы одного цвета (чаще чёрные).

Стандарт ATA всегда устанавливал максимальную длину кабеля равной 46 см. Это ограничение затрудняет присоединение устройств в больших корпусах, или подключение нескольких приводов к одному компьютеру, и почти полностью уничтожает возможность использования дисков PATA в качестве внешних дисков. Хотя в продаже широко распространены кабели большей длины, следует иметь в виду, что они не соответствуют стандарту. То же самое можно сказать и по поводу «круглых» кабелей, которые также широко распространены. Стандарт ATA описывает только плоские кабели с конкретными характеристиками полного и ёмкостного сопротивлений. Это, конечно, не означает, что другие кабели не будут работать, но, в любом случае, к использованию нестандартных кабелей следует относиться с осторожностью.

Если к одному шлейфу подключены два устройства, одно из них обычно называется ведущим (англ. master), а другое ведомым (англ. slave). Обычно ведущее устройство идёт перед ведомым в списке дисков, перечисляемых BIOS’ом компьютера или операционной

системы. В старых BIOS'ах (486 и раньше) диски часто неверно обозначались буквами: «C» для ведущего диска и «D» для ведомого.

Если на шлейфе только один привод, он в большинстве случаев должен быть сконфигурирован как ведущий. Некоторые диски (в частности, производства Western Digital) имеют специальную настройку, именуемую single (т. е. «один диск на кабеле»). Впрочем, в большинстве случаев единственный привод на кабеле может работать и как ведомый (такое часто встречается при подключении CD-ROM'а на отдельный канал).

Настройка, именуемая cable select (т. е., «выбор, определяемый кабелем», кабельная выборка), была описана как опциональная в спецификации ATA-1 и стала широко распространена начиная с ATA-5, поскольку исключает необходимость переставлять перемычки на дисках при любых переподключениях. Если привод установлен в режим cable select, он автоматически устанавливается как ведущий или ведомый в зависимости от своего местоположения на шлейфе. Для обеспечения возможности определения этого местоположения шлейф должен быть с кабельной выборкой. У такого шлейфа контакт 28 (CSEL) не подключен к одному из разъёмов (серого цвета, обычно средний). Контроллер заземляет этот контакт. Если привод видит, что контакт заземлён (то есть на нём логический 0), он устанавливается как ведущий, в противном случае (высокоимпедансное состояние) — как ведомый.

Во времена использования 40-проводных кабелей, широко распространилась практика осуществлять установку cable select путём простого перерезания проводника 28 между между двумя разъёмами, подключаемыми к диску. При этом ведомый привод оказывался на конце кабеля, а ведущий в середине. Такое размещение в поздних версия спецификации было даже стандартизировано. К сожалению, когда на кабеле размещается только одно устройство, такое размещение приводит к появлению ненужного куска кабеля на конце, что нежелательно — как из соображений удобства, так и по физическим параметрам: этот кусок приводит к отражению сигнала, особенно на высоких частотах.

80-проводные кабели, введённые для UDMA4, лишены указанных недостатков. Теперь ведущее устройство всегда находится в конце шлейфа, так что, если подключено только одно устройство, не получается этого ненужного куска кабеля. Кабельная выборка же у них «заводская» — сделанная в самом разъёме просто путём исключения данного контакта. Поскольку для 80-проводных шлейфов в любом случае требовались собственные разъёмы, повсеместное внедрение этого не составило больших проблем. Стандарт также требует использования разъёмов разных цветов, для более простой идентификации их как производителем, так и сборщиком. Синий разъём предназначен для подключения к контроллеру, чёрный — к ведущему устройству, серый — к ведомому.

Термины «ведущий» и «ведомый» были заимствованы из промышленной электроники (где указанный принцип широко используется при взаимодействии узлов и устройств), но в данном случая являются некорректными, и потому не используются в текущей версии стандарта ATA. Более правильно называть ведущий и ведомый диски соответственно device 0 (устройство 0) и device 1 (устройство 1). Существует распространённый миф, что ведущий диск руководит доступом дисков к каналу. На самом деле управление доступом дисков и очерёдностью выполнения команд осуществляют контроллер (которым, в свою очередь, управляет драйвер операционной системы). То есть фактически оба устройства являются ведомыми по отношению к контроллеру.

38. Протокол Serial ATA.

SATA (англ. *Serial ATA*) — последовательный интерфейс обмена данными с накопителями информации (как правило, с жёсткими дисками). SATA является развитием интерфейса ATA (IDE), который после появления SATA был переименован в PATA (Parallel ATA).

Описание SATA

SATA использует 7-контактный разъём вместо 40-контактного разъёма у PATA. SATA-кабель имеет меньшую площадь, за счёт чего уменьшается сопротивление воздуху, обдувающему комплектующие компьютера; улучшается охлаждение системы.

SATA-кабель за счёт своей формы более устойчив к многократному подключению. Питающий шнур SATA так же разработан с учётом многократных подключений. Разъём питания SATA подаёт 3 напряжения питания: +12 В, +5 В и +3,3 В; однако современные устройства могут работать без напряжения +3,3 В, что даёт возможность использовать пассивный переходник со стандартного разъёма питания IDE на SATA. Ряд SATA устройств поставляется с двумя разъёмами питания: SATA и Molex.

Стандарт SATA отказался от традиционного для PATA подключения по два устройства на шлейф; каждому устройству полагается отдельный кабель, что снимает проблему невозможности одновременной работы устройств, находящихся на одном кабеле (и возникавших отсюда задержек), уменьшает возможные проблемы при сборке (проблема конфликта Slave/Master устройств для SATA отсутствует), устраняет возможность ошибок при использовании нетерминированных PATA-шлейфов.

Стандарт SATA предусматривает горячую замену устройств и функцию очереди команд (NCQ).

Разъёмы SATA

SATA устройства используют два разъёма: 7-контактный (подключение шины данных) и 15-контактный (подключение питания). Стандарт SATA предусматривает возможность использовать вместо 15-контактного разъёма питания стандартный 4-контактный разъём Molex. Использование одновременно обоих типов силовых разъёмов может привести к повреждению устройства.

G — заземление (англ. *Ground*)

R — зарезервировано

D1+,**D1−**,**D2+**,**D2−** — два канала передачи данных (от контроллера к устройству и от устройства к контроллеру соответственно). Для передачи сигнала используется технология LVDS, провода каждой пары (D1+, D1− и D2+, D2−) являются экранированными витыми парами.

39.Протокол FCP (Fibre Channel Protocol)

Протокол Fibre Channel предназначен для согласования многочисленных требований, относящихся к возрастающему спросу на высококачественную передачу информации.

Основные задачи Fibre Channel:

- Обеспечение работы множества известных существующих канальных и межсетевых протоколов на одном и том же физическом интерфейсе и в среде передачи
- Обеспечение высокой пропускной способности (100 Мбит/с и выше)
- Гибкие топологии
- Обеспечение возможности соединения на нескольких километрах

- Поддержка разнообразных скоростей передачи данных, типов среды и соединителей

Вообще протокол Fibre Channel пытается объединить в себе оба типа технологий – и канальную, и сетевую. Существует два базовых типа передачи данных между процессорами и между процессорами и периферией: каналы и сети.

Канал представляет собой законченный, непосредственный, структурированный и предсказуемый механизм для передачи данных между относительно немногочисленными объектами. Канал обычно настраивается один раз, не требуя значительного выбора, таким образом обеспечивая высокоскоростную специальную аппаратную среду. Каналы в основном используются для подключения периферийных устройств, таких как дисководы, принтеры, накопители на магнитных лентах и т.д., к рабочей станции. К общепринятым канальным протоколам относится интерфейс малых компьютерных систем (SCSI) и высокоскоростной параллельный интерфейс (HIPPI).

Сети, напротив, являются неструктурированными и непредсказуемыми. Сети способны автоматически подстраиваться под изменяющиеся условия и могут поддерживать значительно большее количество подключаемых узлов. Эти факторы требуют, что для успешной маршрутизации данных из одного пункта в другой требуется более тщательный выбор. Основной выбор осуществляется в программном обеспечении, что делает сети более медлительными, чем каналы.

Примерами общепринятых сетей являются сети Ethernet, Token Ring (кольцевая сеть с маркерным доступом) и FDDI (распределенный интерфейс передачи данных по волоконно-оптическим каналам).

Наиболее вероятно, что протокол Fibre Channel продолжит развитие в сторону рынка запоминающих устройств, который сможет использовать преимущества этого протокола поверх традиционных канальных технологий, таких как SCSI. Способность получения доступа к запоминающим устройствам большой емкости с большей скоростью и на большие расстояния – очень заманчиво для таких приложений, как мультимедиа, медицинское воспроизведение и аналитическая визуализация. Поскольку Fibre Channel обеспечивает передачу данных на большие расстояния, он имеет преимущества в чрезвычайных ситуациях, поскольку запоминающие устройства могут размещаться удаленно.

К сожалению, современная тенденция развития Fibre Channel заключается в том, что продолжается процесс определения все большего количества стандартов, которые усложняют этот протокол. Вероятно это самая большая угроза для будущего этого протокола.

40.Протокол FCIP iFCIP.

В рамках работы над сетевыми технологиями хранения данных в Internet Engineering Task Force (IETF) была создана рабочая группа IP Storage (IPS) по направлениям:
iSCSI (Internet Small Computer Systems Interface)
FCIP (Fibre Channel over TCP/IP)
iFCP (Internet Fibre Channel Protocol)
iSNS (Internet Storage Name Service)

А также, как уже отмечалось, в январе 2001 в рамках SNIA (Storage Networking Industry Association) был организован IP Storage форум. Сегодня форум включает три подгруппы: FCIP, iFCP, iSCSI. Каждая из которых представляет протокол, который находится под протекцией IETF.

FCIP — созданный на базе TCP/IP туннельный протокол, функцией которого является соединение географически отдаленных FC SAN без какого-либо влияния на FC и IP протоколы.

iFCP — созданный на базе TCP/IP протокол для соединения FC систем хранения данных FC сетей хранение данных, используя IP инфраструктуру совместно или вместо FC коммутационных и маршрутизирующих элементов.

Для лучшего понимания позиционирования этих трёх протоколов приведем структурную схему сетей, построенных с их использованием.

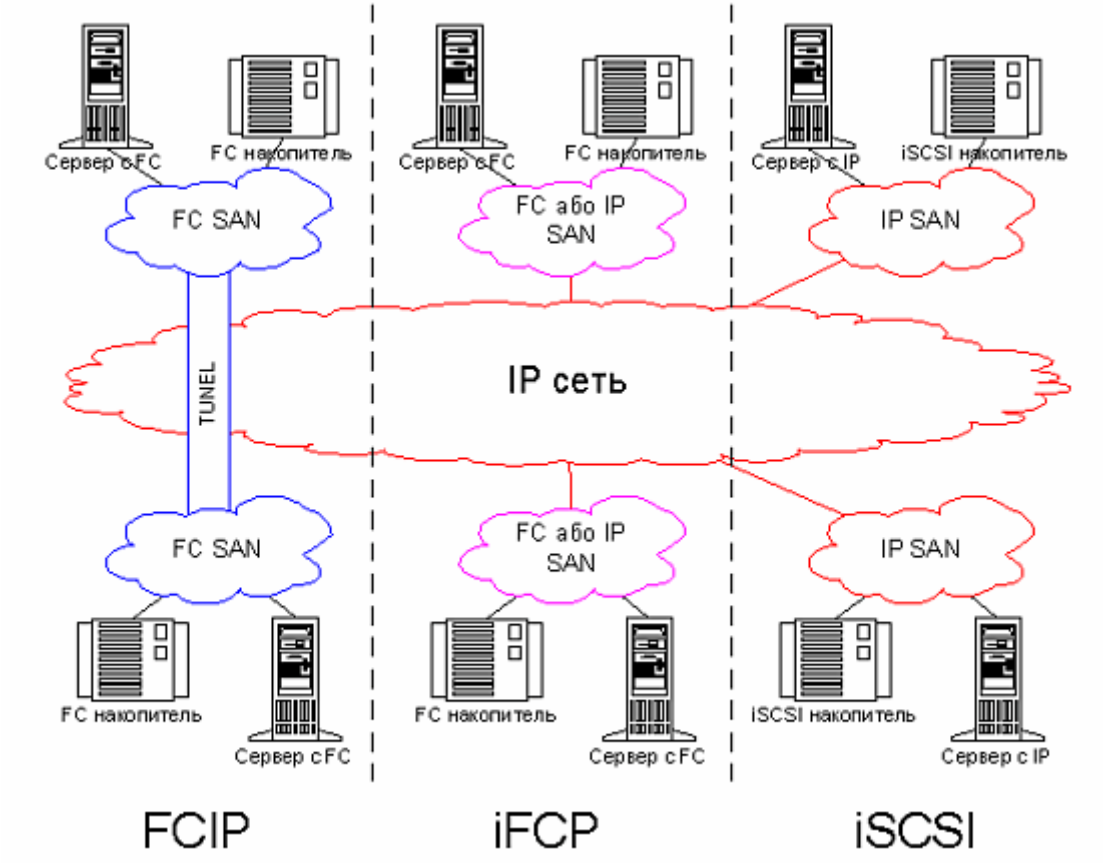


Рисунок 5. Блок-схема IP Storage сетей

Fibre Channel over IP

Наименее революционным из трех названых выше является протокол Fibre Channel over IP. Он не вносит практически никаких изменений в структуру SAN и в организацию самых систем хранения данных. Главная идея этого протокола — реализация возможности объединения географически отдаленных сетей хранения данных.

Вот так выглядит стек протокола FCIP:

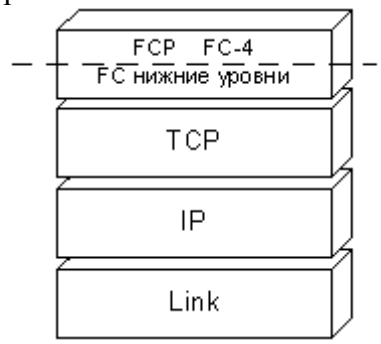


Рисунок 6. Нижние уровни протокола FCIP

FCIP помогает эффективно решить задачу территориального распределения, и объединения SAN на больших расстояниях. Его основными преимуществами является то, что этот протокол полностью прозрачен для существующих FC SAN сетей и ориентирован на использование инфраструктуры современных MAN/WAN сетей. Таким образом, для обеспе-

чения новой функциональности пользователям, которые ищут возможности связать между собою географически отдаленные FC SAN, будет нужен всего лишь FCIP шлюз и подключение к MAN/WAN сети. Географически распределенная SAN, построенная с помощью FCIP, воспринимается SAN устройствами как обычная FC сеть, а для MAN/WAN сети, к которой она подключенная, она представляет обычный IP трафик.

Draft стандарт рабочей группы IETF IPS — FCIP определяет:

- правила инкапсуляции FC кадров для передачи через TCP/IP;
- правила использования инкапсуляции для создания виртуальной связи между FC устройствами и элементами FC сети;
- окружение TCP/IP для поддержки создания виртуальной связи и обеспечение тунелирования FC трафика через IP сеть, включая безопасность, целостность данных и вопрос скорости передачи данных.

Среди прикладных задач, которые можно качественно решить с использованием FCIP протокола: удаленное резервирование, восстановление данных и общий доступ к данным. При использовании высокоскоростных MAN/WAN коммуникаций можно также с успехом применять: синхронное дублирование данных и общий распределенный доступ к системам хранения данных.

iFCP

Internet Fibre Channel Protocol — это протокол, который обеспечивает передачу FC трафика поверх TCP/IP транспорта между шлюзами iFCP. В этом протоколе, транспортный уровень FC замещается транспортом IP сети, трафик между FC устройствами маршрутизируется и коммутируется средствами TCP/IP. Протокол iFCP предоставляет возможность подключать существующие FC системы хранения данных к IP сети с поддержкой сетевых сервисов, которые нужны этим устройствам.

Стек протокола iFCP имеет такой вид:

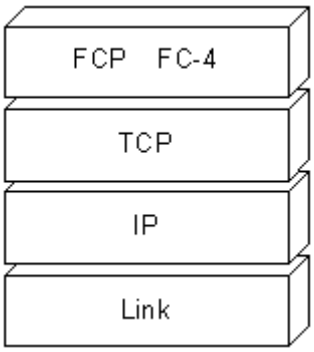


Рисунок 7. Нижние уровни протокола iFCP

iFCP, согласно спецификации:

- накладывает кадры FC для их транспортирования на предварительно определенное TCP соединение;
- FC сервисы передачи сообщений и маршрутизации перекрываются в шлюзовом устройстве iFCP, таким образом, сетевые структуры и компоненты FC не сливаются в общую FC SAN, а управляются средствами TCP/IP;
- динамично создает IP туннели для FC кадров

Важной особенностью iFCP является то, что этот протокол обеспечивает FC device-to-device связь (связь между устройствами) через IP сеть, которая является значительно более гибкой схемой, если сравнивать ее со связью SAN-to-SAN. Так, например, если iFCP имеет

TCP связь между парами портов N_Port двух FC устройств, такая связь может иметь свой собственный уровень QoS, который будет отличаться от уровня QoS другой пары FC устройств.

41. Архитектуры систем хранения данных: Сравнительные характеристики DAS, NAS, SAN, рекомендации по применению.

Типы архитектуры систем хранения данных

Различают три основных вида архитектуры дисковой подсистемы в соответствии с порядком организации доступа к системам хранения: DAS (Direct Attached Storage) – система хранения, непосредственно подключаемая к серверу; NAS (Network Attached Storage) – система хранения, подсоединяемая к сети; SAN (Storage Area Network) – сеть хранения данных. Основой SAN является выделенная специализированная сеть, которая служит исключительно для организации доступа к данным.

Сравнительные характеристики систем хранения данных

Характеристика	NAS	SAS (DAS)	SAN
Протоколы передачи данных	CIFS, HTTP, NFS, FTP	SCSI, SSA	SCSI
Скорость передачи	не менее 100 МБ/с на один порт	несколько сот МБ/с	до 1 Гб/с на один порт
Сетевые протоколы	TCP/IP через Ethernet, FDDI, ATM, Gigabit Ethernet	SCSI-интерфейс сервера, сетевой протокол неприемлем	Fibre Channel, Gigabit Ethernet
Масштабирование	Качественное, но снижает пропускную способность сети	Ограничено количеством подключаемых устройств и производительностью единственного сервера	Самое эффективное
Миграция данных	Используются способы резервирования/ восстановления	Снижает производительность сервера	Обеспечивается построение систем хранения высокой готовности с возможностью дублирования в реальном времени

42.Архитектура систем хранения данных DAS

DAS (DIRECT ATTACHED STORAGE)

Системы хранения с непосредственным подключением к высокоскоростному интерфейсу сервера (Direct Attached Storage) представляют собой традиционный способ оснащения сервера внешней системой хранения данных. Как правило, в качестве такого интерфейса выступает шина SCSI. Основные преимущества DAS-систем – низкая стоимость и высокая скорость обмена данными между системой хранения и сервером. Оборотной стороной этой тесной взаимосвязи являются недостаточная управляемость и неоптимальная утилизация ресурсов. Естественно, для использования DAS-систем необходим сервер, к которому хранилище будет подключено.

В настоящее время DAS-системы занимают лидирующее положениусловлено простотой подключения и относительной дешевизной, однако доля продаж таких систем постоянно уменьшается. В настоящее время компания HP активно продвигает решения DAS с возможностью плавной миграции к SAN (DtS), но это повлечет за собой значительные затраты, поскольку такие программы предлагают физическое перемещение универсальных однодюймовых дисков и контроллеров в новое шасси (причем сохранность информации не всегда гарантируется).
Системы DAS следует использовать при необходимости увеличения дискового пространства одного сервера и вынесения его за корпус. В этом случае ваш сервер должен быть оснащен внешним SCSI-интерфейсом (или FibreChannel). Также можно рекомендовать использовать DAS для рабочих станций, локально обрабатывающих большие объемы информации (например, станций нелинейного видеомонтажа).
Системы DAS не следует использовать, если в ближайшем будущем планируется переход на сеть хранения данных – SAN. В такой ситуации стоит сразу покупать решение SAN начального уровня или решение, которое может быть использовано как в качестве DAS, так и SAN.

43.Архитектура систем хранения данных NAS

NAS (NETWORK ATTACHED STORAGE)
NAS представляет собой готовый файловый сервер с RAID-контроллером, несколькими дисками, поддерживающими механизм «горячей замены», и одним-двумя интерфейсами Gigabit Ethernet. От обычного сервера его отличают собственная операционная система, одновременная поддержка клиентов различных ОС (Windows, Linux, Solaris, Macintosh и т. д.), простота инсталляции. Основное назначение – хранение данных на удаленном носителе, с возможностью разделенного доступа и задания прав пользователей. Доступ к NAS-устройствам осуществляется по локальной сети на уровне протоколов передачи файлов (NFS, CIFS), и со стороны пользователя работа с таким устройством выглядит как подключаемый дисковый сетевой ресурс. NAS снимает такие проблемы традиционного файлового сервера, как дублирование данных и совместное использование файлов.
Основными преимуществами подобного решения по сравнению с обычным классическим файл-сервером являются, пожалуй, лишь экономия на стоимости операционной системы и качество сборки, также не стоит упускать из виду поддержку современных технологий резервного копирования. Еще один плюс сети NAS – она не требует пользовательских лицензий, а также ежедневного администрирования (однако определенную начальную настройку производить так или иначе придется).
NAS следует использовать, когда компании необходимо быстро и без особых хлопот добавить дисковое пространство в локальной сети для клиентов сети. По статистике, NAS обходится на 30% дешевле обычного файлового сервера в сети. Решение на базе NAS поддерживает гетерогенные платформы и оптимизировано под файловый ввод/вывод.
NAS не следует использовать в качестве дискового хранилища для серверов приложений, а также при наличии «узких мест» (bottlenecks) в сети. Не рекомендуется он и для резервного хранения данныхния сервера в ночное время, однако в такой ситуации все преимущества NAS перед обычным файловым сервером теряются.

44.Архитектура систем хранения данных SAN

Сеть хранения данных SAN – это высокоскоростная выделенная сеть передачи данных, связывающая один или несколько серверов с одной или несколькими системами хранения по каналам Fibre Channel. Доступ к данным в SAN осуществляется на уровне блоков (в отличие от NAS, где доступ реализован на уровне файлов). Основная идея SAN состоит в отделении устройств хранения данных от сервера и сетевой ОС. Фактически SAN – это комбинация аппаратных средств и ПО, позволяющая большому числу пользователей хранить и совместно использовать информацию. При использовании SAN сервер не задействует ресурсы на об-

ращение к дискам, а высвобожденные ресурсы направляются на работу выполняемых на нем приложений.
Сеть SAN включает пять базовых компонентов:
• серверы;
• инфраструктуру SAN;
• дисковые системы хранения;
• ленточные системы хранения;
• программное обеспечение управления.
В SAN могут применяться четыре типа топологий:
• «точка – точка» (point-to-point) – прямое подключение сервера к устройству или системе хранения;
• «петля с арбитражным доступом» (Fibre Channel Arbitrated Loop – FC-AL) – передача данных осуществляется последовательно, от узла к узлу (по петле);
• «коммутируемое подключение» (switched, FC-SW) – устройства хранения и серверы (всего до 16 млн единиц) подключаются к коммутатору Fibre Channel.
Полоса пропускания доступна для каждого подключенного устройства. При обращении к внешней памяти допускается одновременное взаимодействие нескольких устройств;
• «смешанное подключение» – используются как коммутаторы, так и концентраторы. К SAN можно подключать серверы разных производителей, использующих разные операционные системы. Количество серверов и операционных систем, которые вам требуются, определяют сложность и стоимость решения SAN.
Одно из самых главных достоинств SAN – простая масштабируемость. Вы можете построить начальную сеть с прямым подключением, затем, при необходимости, добавив к ней коммутатор, подключить еще несколько серверов, ленточный массив для резервного копирования, построить отказоустойчивую фабрику из коммутаторов и по мере укрупнения сети добавлять к ней новые серверы дисковые массивы, внедрять программы управления и автоматической диагностики. Единственный недостаток – достаточно высокая стоимость.
SAN следует использовать, если ваше предприятие применяет или планирует применять ERP, аналитические системы, большие базы данных и другие приложения, для которых характерны высокая дисковая активность и централизованное хранение данных. Также целесообразно применять SAN, если вы хотите подключить к одному дисковому массиву несколько серверов либо расположить серверы и дисковые системы на большом удалении друг от друга.
SAN не следует использовать для организации фай 100 Гб дискового пространства.

45.Представление FCP по уровням модели OSI (физический уровень, кодирование передаваемой информации, контроль канала передачи, сервис передачи данных, FC-4).

Чтобы понять протокол Fibre Channel, нужно понять отдельные части или уровни, из которых строится протокол. Протокол Fibre Channel не придерживается модели взаимосвязи открытых систем (OSI) МСЭ, вместо этого протокол был разбит на пять уровней: FC-0, FC-1, FC-2, FC-3 и FC-4.

Можно примерно сказать, что уровни Fibre Channel определены до транспортного уровня модели OSI. Каждый из этих уровней кратко описан в следующих разделах.

Уровни FC-0 и FC-1 можно сравнить с физическим уровнем модели OSI. Более того, уровень FC-2 похож на другие протоколы, которые определены как уровень управления доступом к среде (MAC), обычно представляемый как нижняя половина уровня звена данных. Однако протокол Fibre Channel не определяет концепции MAC. Уровень FC-3 является весьма неопределенным уровнем, который имеет дело с описанием набора услуг для устройств, имею-

щих больше одного порта. И, наконец, уровень FC-4 определяет, как другие хорошо известные протоколы верхних уровней размещаются и передаются через Fibre Channel. На Рисунке 6 представлены уровни структуры протоколов Fibre Channel.

Уровень FC-0

Самый нижний уровень протокола Fibre Channel, т.е. FC-0, определяет физическое звено системы, включая требования сигнализации, среды передачи (волокно, соединители), а также параметры оптического и электрического приемника и передатчика для множества скоростей передачи данных.

Физический уровень предназначался для использования с самыми разными технологиями, чтобы соответствовать потребностям системы. Поэтому сквозной маршрут передачи данных может включать различные технологии звена данных, чтобы достигнуть максимальной производительности и эффективности согласно задачам технологии Fibre Channel.

Уровень FC-1

Уровень FC-1 определяет протокол передачи, включая правила кодирования и декодирования символов 8B/10B, специальные символы и контроль ошибок. Этот способ кодирования обладает множеством характеристик, упрощающих и удешевляющих конструкцию цепей передатчика и приемника, давая возможность работать с требуемым коэффициентов ошибок по битам (BER), составляющим 10^{-12}. Детали такого способа кодирования описаны в подразделе «Символ передачи 8B/10B» предыдущего раздела «Иерархия передачи».

Уровень FC-2

Протокол сигнализации, т.е. уровень FC-2, служит в качестве транспортного механизма протокола Fibre Channel. Этот уровень определяет правила формирования кадров, управление последовательностями и обменом, управление классами обслуживания, а также управление потоком и входом в систему. Чтобы передавать информацию, этот уровень полагается на иерархию передачи.

Элементы уровня FC-2 также включают кадры управления звеном, механизмы входа в систему и управления потоком, а также классы обслуживания. Кадры управления звеном используются для указания успешного или неуспешного приема каждого кадра данных. Они, в частности, используются для услуг 1-го и 2-го Класса, где другие классы обслуживания, такие как Класс 3, обрабатывают кадр управления звеном выше уровня Fibre Channel. Понятия входа в систему (Login), управления потоком (Flow Control) и классов обслуживания (Classes of Service) описываются в следующих разделах.

Уровень FC-3

Этот уровень обеспечивает протокол формирования кадров и другие услуги, которые управляют операциями на множестве портов в одном узле. Уровень FC-3 находится на стадии разработки, поскольку полные требования для многопортовых функций еще не определены.

Уровень FC-4

Уровень FC-4 определяет размещение протокола Fibre Channel в протокол верхнего уровня (ULP). В настоящее время определено множество типов размещений для множества важных канальных, периферийных и сетевых протоколов:

- Интерфейс малых компьютерных систем (SCSI)
- Высокоскоростной параллельный интерфейс (HIPPI)

- Межсетевой протокол (IP)
- Асинхронный режим переноса – уровень адаптации 5-го типа (ATM-AAL5)
- Интеллектуальный интерфейс периферийных устройств – 3 (IPI-3) (диск и магнитная лента)
- Однобайтные наборы кодов команд (SBCCS) или ESCON/FICON/SBCON

Следует отметить, что тремя самыми важными типами размещения ULP являются протоколы IP, SCSI и FICON.

Сравнение со стеком OSI

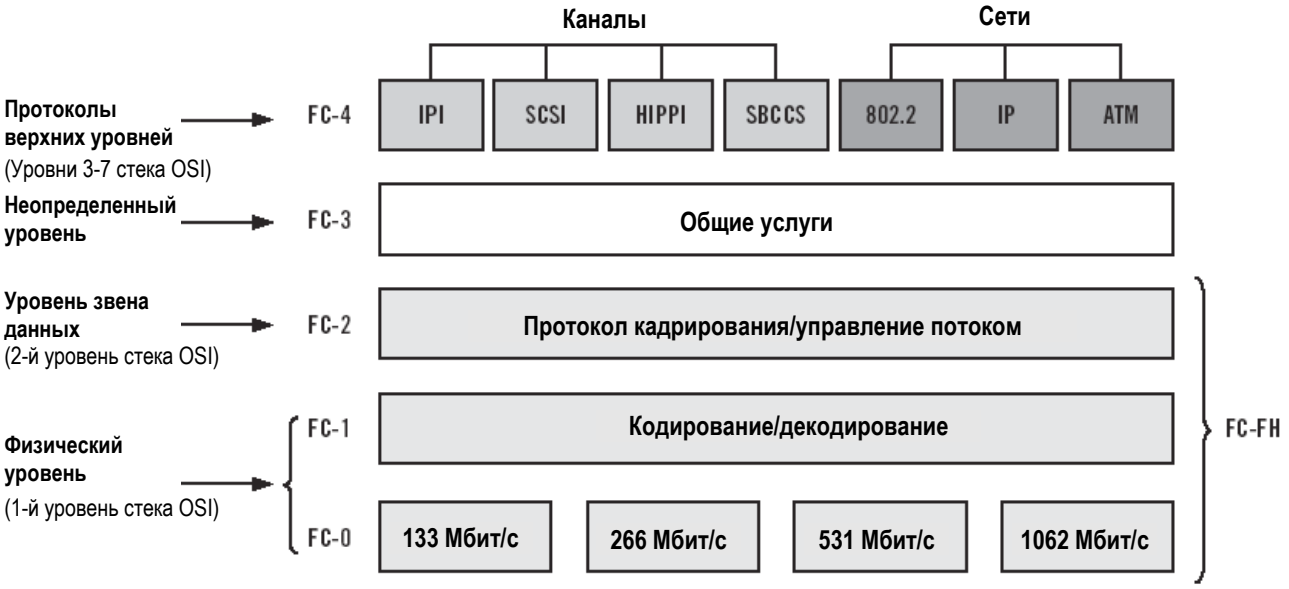


Рисунок 6. Структура протоколов Fibre Channel

46.Типы портов в сетях Fibre Channel.

В зависимости от поддерживаемой топологии и типа устройства порты разделяются на несколько типов:

- Порты узлов:
  - **N_Port** (**Node port**), порт устройства с поддержкой топологии «Точка-Точка».
  - **NL_Port** (**Node Loop port**), порт устройства с поддержкой топологии «Ткань» (Fabric).
- Порты коммутатора/маршрутизатора (только для топологии FC-SW):
  - **F_Port** (**Fabric port**), порт ткани. Используется для подключения портов типа **N_Port** к коммутатору.
  - **FL_Port** (**Fabric Loop port**), порт ткани с поддержкой петли. Используется для подключения портов типа **NL_Port** к коммутатору.
  - **E_Port** (**Expansion port**), порт расширения. Используется для соединения коммутаторов. Может быть соединён только с портом типа **E_Port**.
  - **EX_port**
  - **TE_port** (**Trunking Expansion port**)
- Общий случай:
  - **L_Port** (**Loop port**), порт устройства с поддержкой топологии «Петля».
  - **G_port** (**Generic port**)

47. Понятие о кластеризации: основные определения и термины. Классификация. Сферы применения.

48.Понятие о GRID технологиях.

Термин GRID вычисления (Computing grid) появился по аналогии с термином Power grid (единая энергосистема). Т. е. его можно перевести как единая компьютерная система. Идея очень проста, понятна и давно описана писателями-фантастами. В мире существует множество компьютеров. Давайте объединим их в один большой суперкомпьютер невиданной мощности. Это даст нам огромное количество преимуществ. Сегодня одни компьютеры работают в половину своей мощности, в то время как другие компьютеры перегружены. В то время как в одних странах ночь и компьютеры простаивают, в других странах не хватает вычислительных ресурсов для решения важных и сложных задач. Для некоторых задач (таких как задачи предсказания погоды, моделирование физических процессов, астрофизика и т. д.) необходимы очень мощные компьютеры, которых пока еще не создали. Создание же суперкомпьютера, элементами которого являются обычные компьютеры, принадлежащие различным странам, организациям, людям, позволило бы решить эти проблемы.

Сегодняшняя реальность любой организации такова, что под любое новое коммерческое приложение покупается новый компьютер (компьютеры) и мы имеем множество слабо связанных вычислительных “островков”. Связывание их в единый “континент” даже в рамках одной организации позволило бы резко повысить эффективность использования оборудования и уменьшить количество компьютеров в организации. Имея такой суперкомпьютер неограниченной мощности, любой пользователь может в любое время и в любом месте попросить столько вычислительных ресурсов, сколько ему требуется (и сколько он может оплатить), решить свои задачи и освободить ресурс.

Очень часто в связи с концепцией GRID упоминают термин “computing utility” т е. коммунальная услуга, поскольку GRID позволяет получить вычислительные ресурсы также, как мы получаем другие коммунальные услуги, такие как электричество, газ вода и т д. Когда нам нужно электричество, мы просто находим розетку, включаем прибор и затем оплачиваем по счетчику потребленную электроэнергию. При этом мы не задумываемся о том, на каких ГЭС, ГРЭС, АЭС и т д. электроэнергия была выработана, по каким линиям ЛЭП шла и т д. Концепция GRID позволяет точно также получать и использовать вычислительные ресурсы.

Часто в связи с концепцией GRID также используют термин “виртуализация”. Действительно, в GRID мы работаем не с множеством мелких компьютеров, а с одним виртуальным суперкомпьютером, не с множеством дисков, на которых лежат наши файлы и базы данных, а с единой виртуальной областью хранения данных (огромным виртуальным диском), которая образуется из множества отдельных дисков.

Итак, с точки зрения пользователя GRID не важно, где размещаются данные и какой компьютер будет обрабатывать его запросы. Главное – это то, что пользователь потребовал информацию или выполнение вычислений и получил результат.

1.

3.