Тема 10

Сетевые информационные службы

Содержание темы

- Общие принципы организации сетевых служб.
- Веб-служба.
- Протокол HTTP.
- Почтовая служба.
- Протоколы SMTP, POP3, IMAP.
- Сетевая файловая служба.
- Протокол FTP.
- Служба управления сетью.
- Протоколы SNMP, telnet.

Сетевые службы

Сетевые службы принято делить на несколько групп по типам адресатов предоставляемых ими услуг:

- Службы, ориентированные на конечных пользователей и их приложений (служба печати, файловый сервис, почта, веб-сервис, справочная служба, IP-телефония, служба облачных вычислений).
- Службы, обеспечивающие безопасность сети (сетевая аутентификация, авторизация и контроль доступа).
- Службы, ориентированные на сетевых администраторов, решающих задачи конфигурирования и управления сетевыми устройствами (telnet и SNMP, служба мониторинга и аудита).

Сетевые службы

Сетевые службы принято делить на несколько групп по типам адресатов предоставляемых ими услуг (окончание):

- Службы, помогающие компьютерам и сетевым устройствам предоставлять свои транспортные услуги (служба отображения символьных имен узлов на IP-адреса (DNS) и служба динамического назначения адресов (DHCP)).
- Службы поддержки распределенных вычислений, например, служба репликации, служба вызова удаленных процедур (RPC), являющиеся вспомогательными по отношению к другим службам.

Сетевые службы

Сетевые службы чаще всего представляют собой двухзвенные **распределенные приложения**, одно из звеньев является **клиентом**, другое – **сервером**.

Клиентская и серверная части в общем случае выполняются на разных компьютерах. Как правило, один сервер обслуживает большое число клиентов.

Принципиальной разницей между клиентом и сервером является то, что инициатором выполнения сетевой службой некой работы **всегда выступает клиент**, а сервер всегда находится в режиме пассивного ожидания запросов.

Сетевые службы

Взаимодействие клиента и сервера может выполняться только путем передачи **сообщений** через сеть в соответствии с выбранным **протоколом**.

Основными вопросами разработки распределенных приложений являются:

- распределение функций между его звеньями (клиентом и сервером);
- определение протокола взаимодействия этих звеньев.

Сетевые службы

Протоколы обмена сообщениями, лежащие в основе сетевых служб, относятся к **прикладному уровню**.

Службы, имеющие одно и то же назначение, могут использовать разные протоколы.

Пользовательский интерфейс, который в наше время обычно является графическим (**Graphical User Interface**, **GUI**), - важная часть клиента сетевой службы.

От качества интерфейса зависит удобство работы пользователя с данной реализацией службы (степень дружественности), он также отражает функциональное богатство службы.

Веб-служба

Изобретение в 1989 году Тимом Бернерсом-Ли и Робертом Кайо **Всемирной паутины** (**World Wide Web**, **WWW)** стоит в одном ряду с изобретениями телефона, радио и телевидения.

Миллионы компьютеров, связанных через Интернет, хранят невообразимо огромные объемы информации, представленной в виде веб-страниц.

Веб-страница, или **веб-документ**, как правило, состоит из основного HTML-файла и некоторого количества ссылок на другие объекты разного типа: JPEG- и GIF-изображения, другие HTML-файлы, аудио- и видеофайлы.

Веб-служба

HTML-файлом, HTML-страницей или **гипертекстовой страницей** называют файл, который содержит текст, написанный на **языке HTML** (**HyperText Markup Language** – язык разметки гипертекста).

Браузер находит веб-страницы и отдельные объекты по адресам специального формата, называемым **URL** (**Uniform Resource Locator** – унифицированный указатель ресурса).

URL-адрес может выглядеть, например, так:

http://www.bsut.byk/books/books.htm

В URL-адресе можно выделить три части:

- **Тип протокола доступа.**
- **DNS-имя сервера.**
- **Путь к объекту.**

Веб-служба

Клиентская часть веб-службы, или **веб-клиент,** называемый также **браузером,** представляет собой приложение, которое устанавливается на компьютере конечного пользователя и предназначено для просмотра веб-страниц.

Веб-сервер – это программа, хранящая объекты локально в каталогах компьютера, на котором она запущена, и обеспечивающая доступ к этим объектам по URL-адресам. Наиболее популярными веб-серверами сейчас являются Apache и Microsoft Internet Information Server.

Веб-служба

Как и любой другой сервер, веб-сервер должен быть постоянно в активном состоянии, прослушивая **TCP-порт 80**, являющийся назначенным портом протокола HTTP.

С получением запроса от клиента сервер устанавливает TCP-соединение и получает от клиента имя объекта, после чего находит в своем каталоге этот файл, а также другие связанные с ним объекты, и отсылает их по TCP-соединению клиенту.

Получив объекты от сервера, веб-браузер отображает их на экране.

Веб-служба

Веб-сервер в отношении сеанса с веб-браузером является **сервером без сохранения состояния** (**stateless**).

Это означает, что на сервере не хранится информация, касающаяся состояния сеанса: какие страницы пользователь уже посетил и какие данные ему были переданы.

Такой режим общения с клиентом упрощает организацию сервера, которому необходимо отвечать на большой поток запросов различных пользователей, так что запоминание состояния сеансов пользователей существенно увеличило бы нагрузку на веб-сервер.

Протокол HTTP

HTTP (**HyperText Transfer Protocol** – протокол передачи гипертекста) – это протокол прикладного уровня.

Обмен сообщениями идет по обычной схеме «запрос-ответ».

Клиент и сервер обмениваются **текстовыми** сообщениями стандартного формата, то есть каждое сообщение представляет собой несколько строк обычного текста в кодировке ASCII.

Для транспортировки HTTP-сообщений служит протокол TCP.

Протокол HTTP

При этом TCP-соединения могут использоваться двумя разными способами:

- **Долговременное соединение** – передача в одном TCP-соединении нескольких объектов, причем время существования соединения определяется при конфигурировании вебслужбы.
- **Кратковременное соединение** – передача в рамках одного TCP-соединения только одного объекта.

Протокол HTTP

Долговременное соединение, в свою очередь, может быть использовано двумя способами:

- **Последовательная передача запросов с простоями** – это способ, посредством которого новый запрос посылается только после получения ответа.
- **Конвейерная передача** – это более эффективный способ, в котором следующий запрос посылается до прибытия ответа на один или несколько предыдущих запросов (напоминает метод скользящего окна).

Протокол HTTP

Форматы стартовых строк и заголовков HTTP

Обобщенная структура сообщения	**HTTP-запрос**	**HTTP-ответ**
Стартовая строка (всегда должна быть первой строкой сообщения; обязательный элемент)	Формат запроса Метод/ URL HTTP/1.x. Пример: GET /books/books.htm HTTP/1.1	Формат ответа: HTTP/1.x КодСостояния Фраза. Пример: HTTP/1.1 200 OK
Заголовки (следуют в произвольном порядке; могут отсутствовать)	Заголовок о DNS-имени компьютера, на котором расположен веб-сервер. Пример: Host: www.olifer.co.uk	Заголовок о времени отправления данного ответа. Пример: Date: 1 Jan 2009 14:00:30
	Заголовок об используемом браузере. Пример: User-agent: Mozilla/5.0	Заголовок об используемом веб-сервере. Пример: Server: Apache/1.3.0 (Unix)
	Заголовок о предпочтительном языке. Пример: Accept-language: ru	Заголовок о количестве байтов в теле сообщения. Пример: Content-Length: 1234
	Заголовок о режиме соединения. Пример: Connection: close	Заголовок о режиме соединения. Пример: Connection: close
Пустая строка		
Тело сообщения (может отсутствовать)	Здесь могут быть расположены ключевые слова для поисковой машины или страницы для передачи на сервер	Здесь может быть расположен текст запрашиваемой страницы

Протокол HTTP

Стартовая строка запроса включает в себя поле **метода** – это название операции, которая должна быть выполнена.

Чаще всего в запросах используется метод **GET**, то есть запрос объекта.

Метод **HEAD** аналогичен методу GET, но запрашиваются только метаданные заголовка HTML-страницы.

Метод **POST** используется клиентом для отправки данных на сервер: сообщений электронной почты, ключевых слов в запросе поиска, веб-формы.

Метод **PUT** используется клиентом для размещения некоторого объекта на сервере, на который указывает URL-адрес.

Метод **DELETE** указывает серверу на то, что некоторый объект на сервере, **определяемый URL-адресом, необходимо удалить.**

Почтовая служба

Сетевая почтовая служба, или **электронная почта**, **-** это распределенное приложение, главной функцией которого является предоставление пользователям сети возможности обмениваться электронными сообщениями.

Электронная почта построена в архитектуре клиент-сервер. Почтовый клиент всегда располагается на компьютере пользователя, а почтовый сервер, как правило, работает на выделенном компьютере.

Почтовая служба

Почтовый клиент (называемый также **агентом пользователя) –** это программа, предназначенная для поддержания пользовательского интерфейса (обычно графического), а также для предоставления пользователю широкого набора услуг по подготовке электронных сообщений.

В число таких услуг входит создание текста в различных форматах и кодировках, сохранение, уничтожение, переадресация, сортировка писем по разным критериям, просмотр перечня поступивших и отправленных писем, грамматическая и синтаксическая проверка текста сообщений, ведение адресных баз данных, автоответы, образование групп рассылки и прочее, и прочее.

Почтовая служба

Почтовый сервер выполняет прием сообщений от клиентов, для чего постоянно находится в активном состоянии.

Кроме того, он выполняет буферизацию сообщений, распределение поступивших сообщений по индивидуальным буферам (почтовым ящикам) клиентов, управляет объемами памяти, выделяемой клиентам, выполняет регистрацию клиентов и регламентирует их права доступа к сообщениям, а также решает много других задач.

Почтовая служба

Почтовая служба оперирует **электронными сообщениями –** информационными структурами определенного стандартного формата.

Упрощенно электронное сообщение может быть представлено в виде двух частей, одна из которых (**заголовок**) содержит вспомогательную информацию для почтовой службы, а другая (**тело сообщения**) – это собственно то «письмо», которое предназначается для прочтения, прослушивания или просмотра адресатом (RFC 822).

Почтовая служба

Главными элементами заголовка являются адреса отправителя и получателя в виде

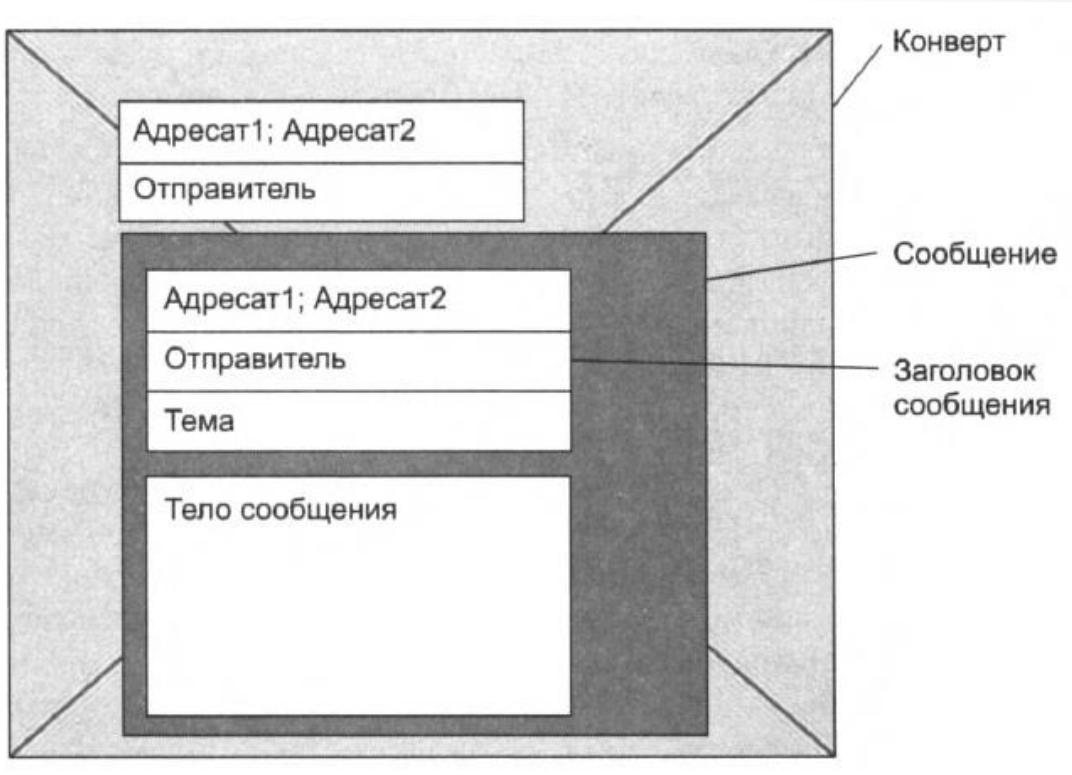
pashabuoy@rambler.ru

где pashabuoy – идентификатор пользователя почтовой службы;

rambler.ru – имя домена, к которому относится этот пользователь.

Почтовая служба

При транспортировке через Интернет почтовое сообщение помещается в **конверт** (**envelope**), который также имеет несколько служебных полей, например поле отправителя и поля получателей.



Почтовая служба

Важную роль в расширении возможности электронной почты по передаче мультимедийной информации сыграл стандарт **MIME** (**Multipurpose Internet Mail Extensions – многоцелевые расширения почты Интернета**).

Этот стандарт описывает структуру сообщения, состоящего из нескольких частей, каждая из которых имеет свои заголовок и тело.

Заголовок описывает тип данных, которые содержатся в теле.

Почтовая служба

Типы данных сообщения:

- обычные текстовые данные в формате ASCII;
- текст в 8-битном формате (такая возможность описана в документе RFC 6152, принятом в марте 2011 года);
- текст не в формате ASCII, преобразованный в ASCII-код (например, с помощью алгоритма base64);
- гипертекст (HTML);
- изображение;
- видеоклип;
- звуковой файл.

Почтовая служба

Части отделяются друг от друга последовательностью символов, называемой **границей** (**boundary**).

Граница не должна встречаться в теле частей сообщения.

Одна из спецификаций стандарта MIME (RFC 1847) относится к расширениям безопасности, поэтому этот документ называют спецификацией **S/MIME** (**Security MIME**).

В S/MIME описаны два новых типа частей MIME:

- **цифровая подпись** (**Mutipart/Signed**);
- **шифрованное тело** (**Multipart/Encrypted**).

Протокол SMTP

В качестве средств передачи сообщения почтовая служба Интернета использует стандартный, разработанный специально для почтовых систем протокол **SMTP** (**Simple Mail Transfer Protocol – простой протокол передачи почты**).

Этот протокол является одним из первых стандартизованных протоколов прикладного уровня (август 1982 года).

SMTP реализуется несимметричными взаимодействующими частями: SMTP-клиентом, работающим на стороне отправителя, и SMTP-сервером, работающим на стороне получателя.

SMTP-сервер должен постоянно быть в режиме подключения, ожидая запросов со стороны SMTP-клиента.

Протокол SMTP

Логика протокола SMTP:

1. После того как, применяя графический интерфейс своего почтового клиента, пользователь щелкает на значке отправки сообщения, SMTP-клиент посылает запрос на установление TCP-соединения на порт 25 SMTP-сервера (это назначенный порт).

Протокол SMTP

Логика протокола SMTP:

2. Если сервер готов, то он посылает свои идентификационные данные, в частности свое DNS-имя.

Если SMTP-сервер оказался не готов, то он посылает соответствующее сообщение клиенту, и тот снова посылает запрос, пытаясь заново установить соединение.

Затем клиент передает серверу почтовые адреса (имена) отправителя и получателя.

Если имя получателя соответствует ожидаемому, то после получения адресов сервер дает согласие на установление SMTP-соединения, и в рамках этого логического канала происходит передача сообщения.

Протокол SMTP

Логика протокола SMTP:

3. Если после приема тела сообщения сервер отвечает командой OK, это означает, что сервер принял на себя ответственность по дальнейшей передаче сообщения получателю.

Однако это не означает, что сервер гарантирует успешную доставку, потому что последнее зависит не только от него: например, клиентская машина получателя может быть в течение длительного времени не подсоединена к Интернету.

Если сервер не может доставить сообщение, то он передает отчет об ошибке отправителю сообщения и разрывает соединение.

Протокол SMTP

Используя одно TCP-соединение, клиент может передать несколько сообщений, предваряя каждое из них указанием почтовых адресов отправителя и получателя.

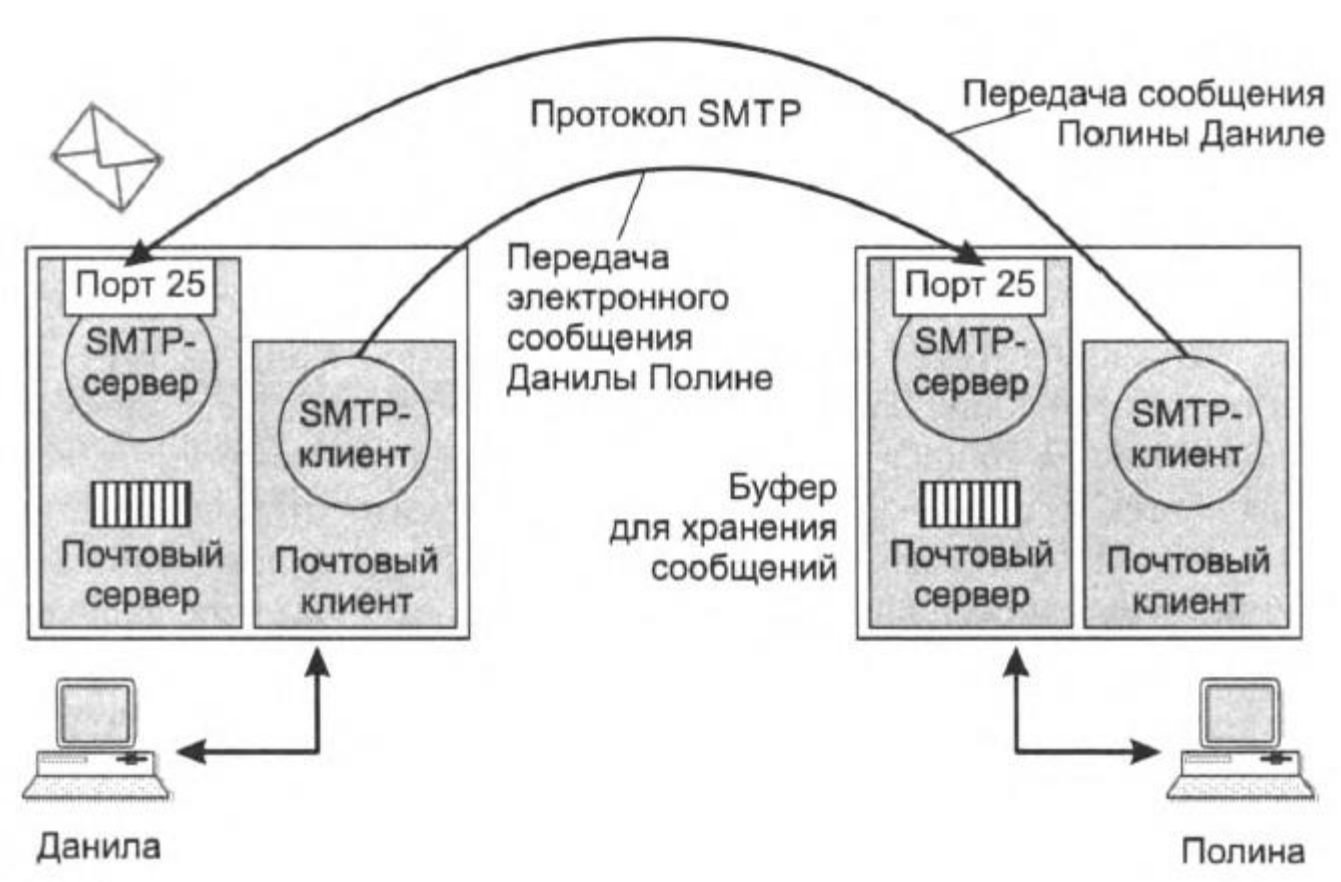
После завершения передачи сообщения TCP- и SMTP-соединения разрываются, и благополучно переданное сообщение сохраняется в буфере на сервере.

В протоколе SMTP предусмотрены как положительные, так и отрицательные уведомления о доставке (промежуточной или окончательной) электронного письма.

Однако только отрицательные уведомления являются обязательными.

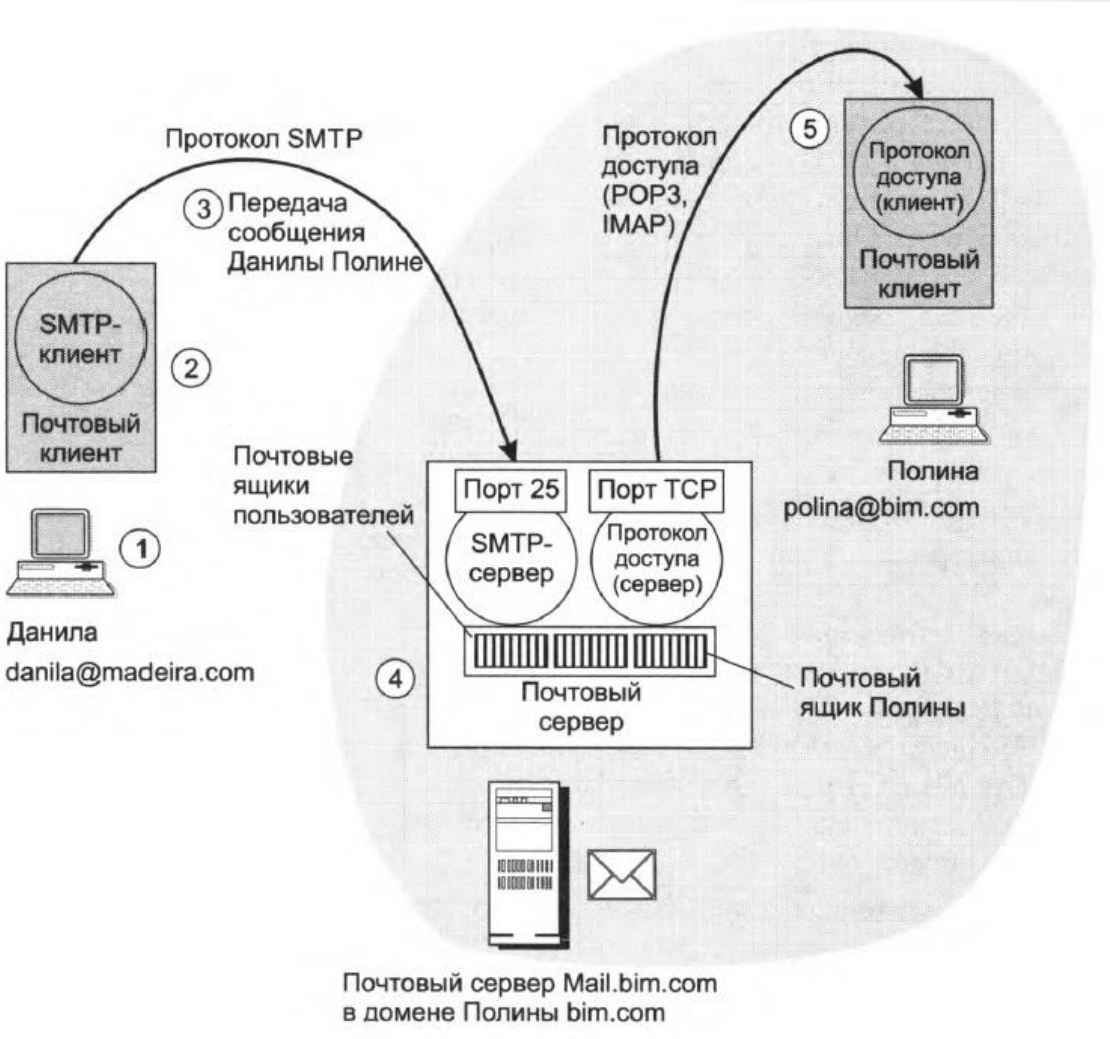
Протокол SMTP

Непосредственное взаимодействие клиента и сервера



Протокол SMTP

Схема с выделенным почтовым сервером



Протокол SMTP

1. Данила решает послать письмо Полине и запускает на своем компьютере программу почтового клиента.

Он пишет текст сообщения, указывает необходимую сопроводительную информацию и отправляет сообщение.

Клиент обращается к системе DNS, чтобы определить имя почтового сервера, обслуживающего домен Полины bim.com.

Получив от DNS в качестве ответа имя mail.bim.com, SMTP-клиент еще раз обращается к DNS – на этот раз чтобы узнать IP-адрес почтового сервера mail.bim.com.

2. SMTP-клиент посылает по данному IP-адресу запрос на установление TCP-соединения через порт 25 (SMTP-сервер).

Протокол SMTP

3. С этого момента начинается диалог между клиентом и сервером по протоколу SMTP.

Направление передачи запроса от клиента на установление SMTP-соединения совпадает с направлением передачи сообщения.

Если сервер оказывается готовым, то после установления TCP-соединения сообщение Данилы передается.

Протокол SMTP

4. Письмо сохраняется в буфере почтового сервера, а затем направляется в индивидуальный буфер, отведенный системой для хранения корреспонденции Полины. Такого рода буферы называют почтовыми ящиками. Важно заметить, что, помимо Полины, у почтового сервера имеется и много других клиентов, что усложняет его работу.

Таким образом, почтовый сервер должен решать самые разнообразные задачи по организации многопользовательского доступа, включая управление разделяемыми ресурсами и обеспечение безопасного доступа.

Протокол SMTP

5. В какой-то момент, который принципиально не связан с моментом поступления сообщений на почтовый сервер, Полина запускает свою почтовую программу и выполняет команду проверки почты.

После этой команды почтовый клиент должен запустить протокол доступа к почтовому серверу.

Однако это не будет SMTP. Протокол SMTP используется тогда, когда необходимо передать данные на сервер, а Полине, напротив, нужно получить их с сервера.

Для этого случая были разработаны другие протоколы, обобщенно называемые протоколами доступа к почтовому серверу, такие, например, как **POP3** и **IMAP.**

Протокол SMTP

Протоколы **POP3** и **IMAP** относятся к протоколам, ориентированным на прием данных. Инициатором передачи сообщений от почтового сервера почтовому клиенту по протоколу POP3 или IMAP является клиент.

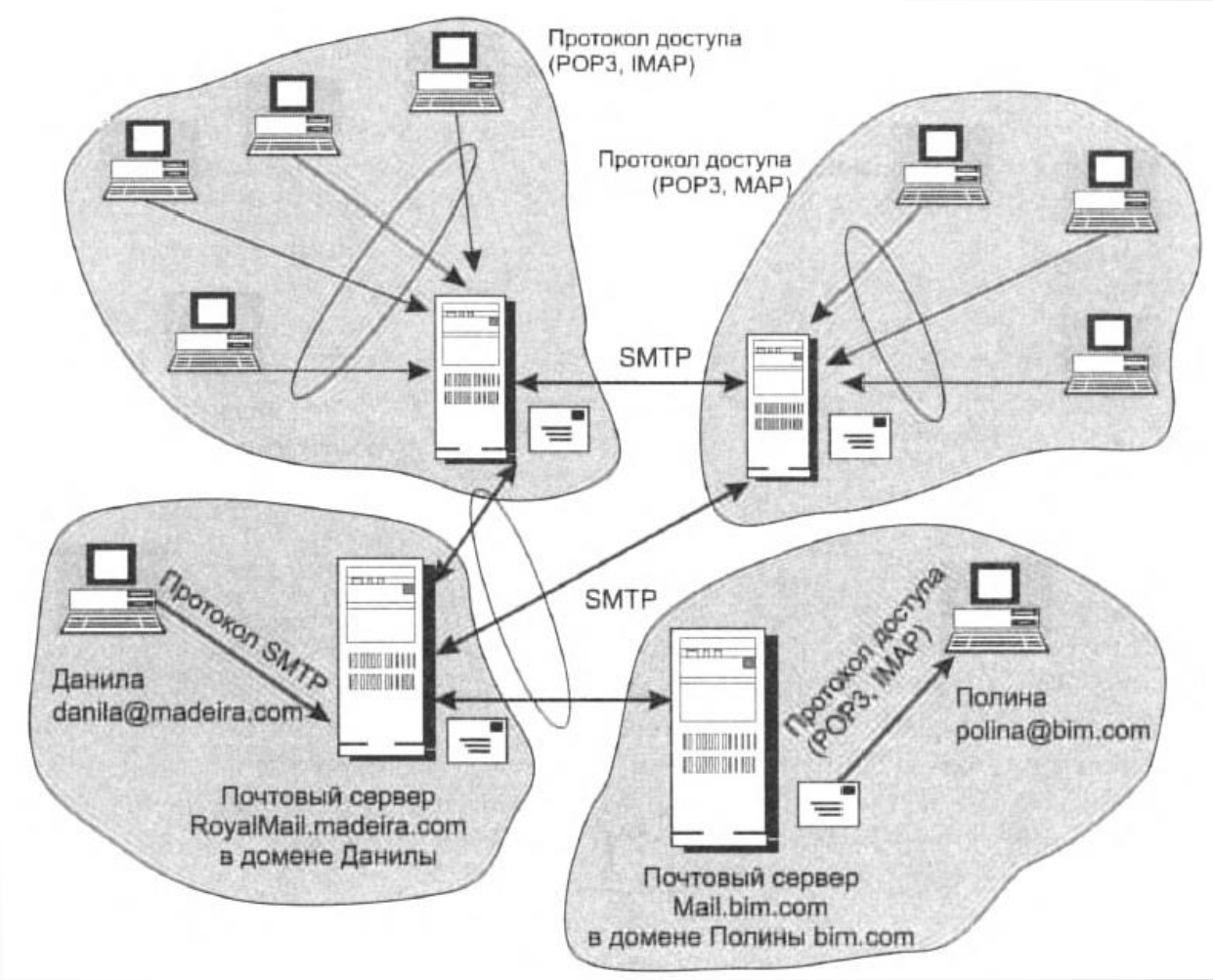
Почтовый сервер ожидает запрос на установление TCP-соединения по протоколу POP3 через порт 110, а по протоколу IMAP – через порт 143.

В результате работы любого из них письмо Данилы оказывается в памяти компьютера Полины.

Направление запроса от клиента к серверу не совпадает с направлением передачи данных, показанному стрелкой.

Протокол SMTP

Схема с двумя почтовыми серверами-посредниками



Протокол POP3

POP3 (Post Office Protocol v.3 – протокол почтового отделения версии 3).

Получая доступ к почтовому серверу по протоколу POP3, вы «перекачиваете» адресованные вам сообщения в память своего компьютера, при этом на сервере не остается никакого следа от считанной вами почты.

Проблемы POP3:

1. При работе на 2-х компьютерах одно письмо можно прочитать только на одном из компьютеров.
2. Клиент не может пропустить, не читая, ни одного письма, поступающего от сервера при работе на одном компьютере.

Протокол IMAP

IMAP (Internet Mail Access Protocol – протокол доступа к электронной почте Интернета).

Если доступ осуществляется по протоколу IMAP, то в память вашего компьютера передаются только копии сообщений, хранящихся на почтовом сервере.

Протокол IMAP

Вся совокупность полученной корреспонденции останется в полной сохранности в памяти почтового сервера (если, конечно, не поступит специальной команды от пользователя об удалении того или иного письма).

Такая схема доступа делает возможным для сервера предоставление широкого перечня услуг по рациональному ведению корреспонденции, то есть именно того, чего лишен пользователь при применении протокола POP3.

Важным преимуществом IMAP является также возможность предварительного чтения заголовка письма, после чего пользователь может принять решение о том, есть ли смысл получать с почтового сервера само письмо.

Сетевая файловая служба

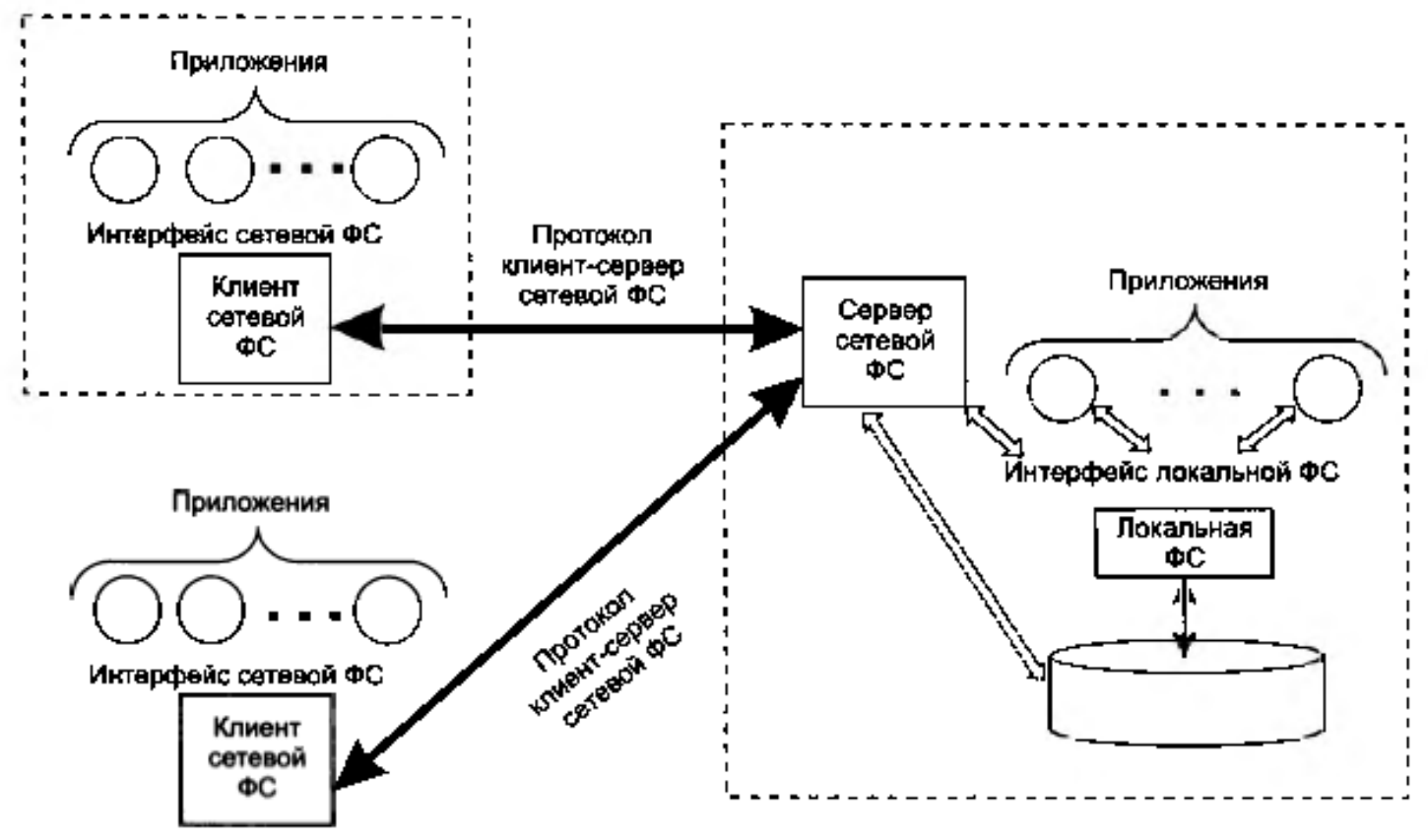
Сетевая файловая служба, или **система (ФС)**, предоставляет пользователям сети услуги по совместному использованию файлов, хранящихся на компьютерах сети.

Сетевая ФС в общем случае включает следующие элементы:

- клиент сетевой ФС;
- сервер сетевой ФС;
- интерфейс сетевой ФС;
- локальная ФС;
- интерфейс локальной ФС;
- протокол клиент-сервер сетевой ФС.

Сетевая файловая служба

Схема сетевой файловой системы



Сетевая файловая служба

Клиенты сетевой ФС установлены на многочисленных компьютерах, подключенных к сети.

Они обслуживают запросы приложений на доступ к файлам, хранящимся на удаленном компьютере.

Клиент сетевой ФС передает по сети запросы серверу сетевой ФС, работающему на удаленном компьютере.

Сетевая файловая служба

Сервер, получив запрос, может обслужить его либо самостоятельно, либо, что является более распространенным вариантом, передать запрос для обслуживания локальной файловой системе.

После получения ответа от локальной файловой системы сервер передает его по сети клиенту, а тот, в свою очередь, - приложению, обратившемуся с запросом.

Сетевая файловая служба

Приложения обращаются к клиенту сетевой ФС, используя определенный программный интерфейс, который в данном случае является **интерфейсом сетевой ФС.**

Этот интерфейс стараются сделать как можно более похожим на **интерфейс локальной ФС**, чтобы соблюсти принцип прозрачности.

Сетевая файловая служба

Клиент и сервер взаимодействуют друг с другом через сеть по **протоколу сетевой файловой системы.**

Если интерфейсы локальной и сетевой файловых систем совпадают, этот протокол может быть достаточно простым – в его функции должна входить ретрансляция серверу запросов, принятых клиентом от приложений, с которыми тот затем будет обращаться к локальной ФС.

Сетевая файловая служба

Протокол сетевой ФС, помимо простой ретрансляции системных файловых вызовов от клиента серверу, может выполнять и более сложные функции, учитывающие природу сетевого взаимодействия, например то, что клиент и сервер работают на разных компьютерах, которые могут оказаться неработоспособными, или что сообщения передаются по ненадежной и вносящей порой большие задержки сетевой среде.

Сетевая файловая служба

Локальная файловая система **FAT** и клиент-сервер протокол **SMB** (**Server Message Block**).

Расширенные версии протокола **SMB** получили название **CIFS** (**Common Internet File System**).

Протокол SMB/ CIFS является основой сетевой файловой службы в операционных системах семейства Windows компании Microsoft.

Сетевая файловая служба

Работа протокола начинается с того, что клиент отправляет серверу специальное сообщение с запросом на установление соединения.

В процессе установления соединения SMB-клиент и SMB-сервер обмениваются информацией о себе: они сообщают друг другу, какой диалект протокола SMB они будут использовать в этом соединении (диалект здесь – определенное подмножество функций протокола, так как помимо файловых функций SMB поддерживает доступ к принтерам, управление внешними устройствами и некоторые другие).

Сетевая файловая служба

Если сервер готов к установлению соединения, он отвечает сообщением-подтверждением.

После установления соединения клиент может обращаться к серверу, передавая ему в SMB-сообщениях команды манипулирования файлами и каталогами.

Клиент может запросить сервер создать и удалить каталог, предоставить содержимое каталога, создать и удалить файл, прочитать и записать содержимое файла, установить атрибуты файла и т. п.

Сетевая файловая служба

В среде операционной системы **Unix** наибольшее распространение получили две сетевые файловые системы и соответственно два протокола клиент-сервер – **FTP** (**File Transfer Protocol**) и **NFS** (**Network File System**).

Протокол FTP

Сетевая ФС на основе протокола **FTP** представляет собой одну из наиболее ранних служб доступа к удаленным файлам.

Первые спецификации FTP относятся к 1971 году.

FTP-серверы и FTP-клиенты имеются практически в каждой ОС семейства **Unix**, а также во многих других сетевых ОС.

FTP-клиенты встроены сегодня в программы просмотра (браузеры) Интернета, так как архивы файлов на основе протокола FTP по-прежнему популярны, и для доступа к таким архивам браузер использует протокол FTP.

Протокол FTP

Протокол FTP целиком перемещает файл с удаленного компьютера на локальный и наоборот, то есть работает по схеме загрузки-выгрузки.

Кроме того, он поддерживает несколько команд просмотра удаленного каталога и перемещения по каталогам удаленной файловой системы.

В протокол FTP встроены примитивные средства аутентификации удаленных пользователей на основе передачи по сети пароля в открытом виде.

Кроме того, поддерживается анонимный доступ (является более безопасным, так как не подвергает пароли угрозе перехвата).

Протокол FTP

Протокол FTP выполнен по схеме клиент-сервер.

FTP-клиент включает три функциональных модуля:

- **User Interface** – модуль, поддерживающий интерфейс клиента с пользователем (принимает от пользователя символьные команды и воспроизводит состояние FTP-сеанса на символьном экране).
- **User-PI** – интерпретатор команд пользователя (взаимодействует с соответствующим модулем FTP-сервера).
- **User-DTP** – модуль, осуществляющий передачу данных файла по командам, получаемым от модуля User-PI по протоколу клиент-сервер (взаимодействует с локальной файловой системой клиента).

Протокол FTP

Символьные клиенты обычно поддерживают следующий основной набор команд:

- **open *имя_хоста* –** открытие сеанса с удаленным сервером;
- **bye** – завершение сеанса с удаленным хостом и завершение работы утилиты ftp;
- **close** – завершение сеанса с удаленным хостом, утилита ftp продолжает работать;
- **ls** (**dir**) – печать содержимого текущего удаленного каталога
- **get *имя_файла*** – копирование удаленного файла на локальный хост;
- **put *имя_файла*** – копирование локального файла на удаленный сервер.

Протокол FTP

FTP-сервер включает два модуля^

- **Server-PI** – модуль, который принимает и интерпретирует команды, передаваемые по сети модулем User-PI.
- **Server-DTP** – модуль, управляющий передачей данных файла по командам от модуля Server-PI (дзаимодействует с локальной файловой системой сервера).

Протокол FTP

FTP-клиент и FTP-сервер поддерживают параллельно два сеанса – **управляющий сеанс** и **сеанс передачи данных**.

Управляющий сеанс открывается при установлении первоначального FTP-соединения клиента с сервером, причем в течение одного управляющего сеанса может последовательно проходить несколько сеансов передачи данных, в рамках которых передается или принимается несколько файлов.

Протокол FTP

Общая схема взаимодействия клиента и сервера выглядит следующим образом.

1. FTP-сервер всегда открывает управляющий TCP-порт 21 для прослушивания, ожидая прихода запроса на установление управляющего FTP-сеанса от удаленного клиента.

2. После установления управляющего соединения клиент отправляет на сервер команды, которые уточняют параметры соединения: имя и пароль клиента, роль участников соединения (активная или пассивная), порт передачи данных, тип передачи, тип передаваемых данных (двоичные данные или ASCII-код), директивы на выполнение действий (читать файл, писать файл, удалить файл и т. п.).

Протокол FTP

3. После согласования параметров пассивный участник соединения переходит в режим ожидания открытия соединения на порт передачи данных.

Активный участник инициирует открытие соединения и начинает передачу данных.

4. После окончания передачи данных соединение по портам данных закрывается, а управляющее соединение остается открытым.

Пользователь может по управляющему соединению активизировать новый сеанс передачи данных.

Протокол FTP

В протоколе FTP предусмотрены специальные команды для взаимодействия FTP-клиента с FTP-сервером (не путать с командами пользовательского интерфейса клиента).

Эти команды делятся на три группы:

1 **Команды управления доступом к системе** доставляют серверу имя и пароль клиента, изменяют текущий каталог на сервере, повторно инициализируют, а также завершают управляющий сеанс.

Протокол FTP

2 **Команды управления потоком данных** устанавливают параметры передачи данных.

Служба FTP может применяться для передачи разных типов данных (ASCII-код или двоичные данные), работать как со структурированными данными (файл, запись, страница), так и с неструктурированными.

Протокол FTP

3 **Команды службы FTP** управляют передачей файлов, операциями над удаленными файлами и каталогами.

Например, команды **RETR** и **STOR** запрашивают передачу файла соответственно от сервера на клиентский хост и наоборот.

Параметрами каждой из этих команд является имя файла.

Команды **DELE, MKD, RMD, LIST** соответственно удаляют файл, создают каталог, удаляют каталог и передают список файлов текущего каталога.

Каждая команда протокола FTP передается в виде одной строки ASCII-кода.

Служба управления сетью

Система управления сетью (**Network Management System**, **NMS) –** это сложный программно-аппаратный комплекс, который контролирует сетевой трафик и управляет коммуникационным оборудованием крупной компьютерной сети.

Системы управления сетью работают, как правило, в **автоматизированном** режиме, выполняя наиболее простые действия автоматически и оставляя человеку принятие сложных решений на основе подготовленной системой информации.

Служба управления сетью

Система управления сетью предназначена для решения следующих групп задач:

- **Управление конфигурацией сети и именованием** заключается в конфигурировании параметров как отдельных элементов сети, так и сети в целом.
- **Обработка ошибок** включает выявление, определение и устранение последствий сбоев и отказов.
- **Анализ производительности и надежности** связан с оценкой на основе накопленной статистической информации таких параметров, как время реакции системы, пропускная способность реального или виртуального канала связи между двумя конечными абонентами сети, интенсивность трафика в отдельных сегментах и каналах сети.

Служба управления сетью

Система управления сетью предназначена для решения следующих групп задач (окончание):

- **Управление безопасностью** подразумевает контроль доступа к ресурсам сети и сохранение целостности данных при их хранении и передаче через сеть.
- **Учет работы сети** включает регистрацию времени использования различных ресурсов сети (устройств, каналов и транспортных служб) и ведение биллинговых операций (плата за ресурсы).

Служба управления сетью

В тех случаях, когда управляемыми объектами являются компьютеры, а также их системное и прикладное программное обеспечение, для системы управления часто используют особое название – **система управления системой** (**System Management System**, **SMS**).

SMS обычно автоматически собирает информацию об установленных в сети компьютерах и создает записи в специальной БД об аппаратных и программных ресурсах.

Служба управления сетью

SMS может централизованно устанавливать и администрировать приложения, которые запускаются с серверов, а также удаленно измерять наиболее важные параметры компьютера, ОС, СУБД (например, коэффициент использования процессора или физической памяти, интенсивность страничных прерываний и др.).

SMS позволяет администратору брать на себя удаленное управление компьютером в режиме эмуляции графического интерфейса популярных операционных систем.

Служба управления сетью

Для решения перечисленных выше задач необходимо иметь возможность управления отдельным устройством (объектом).

Обычно каждое устройство, требующее достаточно сложного конфигурирования, производитель сопровождает автономной программой конфигурирования и управления, работающей в среде специализированной ОС, установленной на этом устройстве – **агентом.**

Служба управления сетью

Агенты могут встраиваться в управляемое оборудование либо работать на устройстве, подключенном к интерфейсу управления такого устройства.

Один агент в общем случае может управлять несколькими однотипными устройствами, поддерживая интерфейс с оператором/администратором, который посылает ему запросы и команды на выполнение определенных операций.

Служба управления сетью

Агент может выполнять следующие функции:

- хранить, извлекать и передавать по запросам извне информацию о технических и конфигурационных параметрах устройства, включая модель устройства, число портов, тип портов, тип ОС, связи с другими устройствами и др.;
- выполнять, хранить и передавать по запросу извне измерения (подсчеты) характеристик функционирования устройства: число принятых пакетов, число отброшенных пакетов, степень заполнения буфера, состояние порта (рабочее или нерабочее);
- изменять по командам, полученным извне, конфигурационные параметры.

Служба управления сетью

В описанной схеме агент играет роль **сервера**, к которому обращается **клиент-администратор** с запросами о значениях характеристик или об установлении конфигурационных параметров управляемого устройства.

Для получения требуемых данных об объекте, выдачи на него управляющих воздействий агент должен иметь возможность взаимодействовать с ним.

Многообразие типов управляемых объектов не позволяет стандартизовать способ взаимодействия агента с объектом.

Эта задача решается разработчиками при встраивании агентов в коммуникационное оборудование или в ОС.

Служба управления сетью

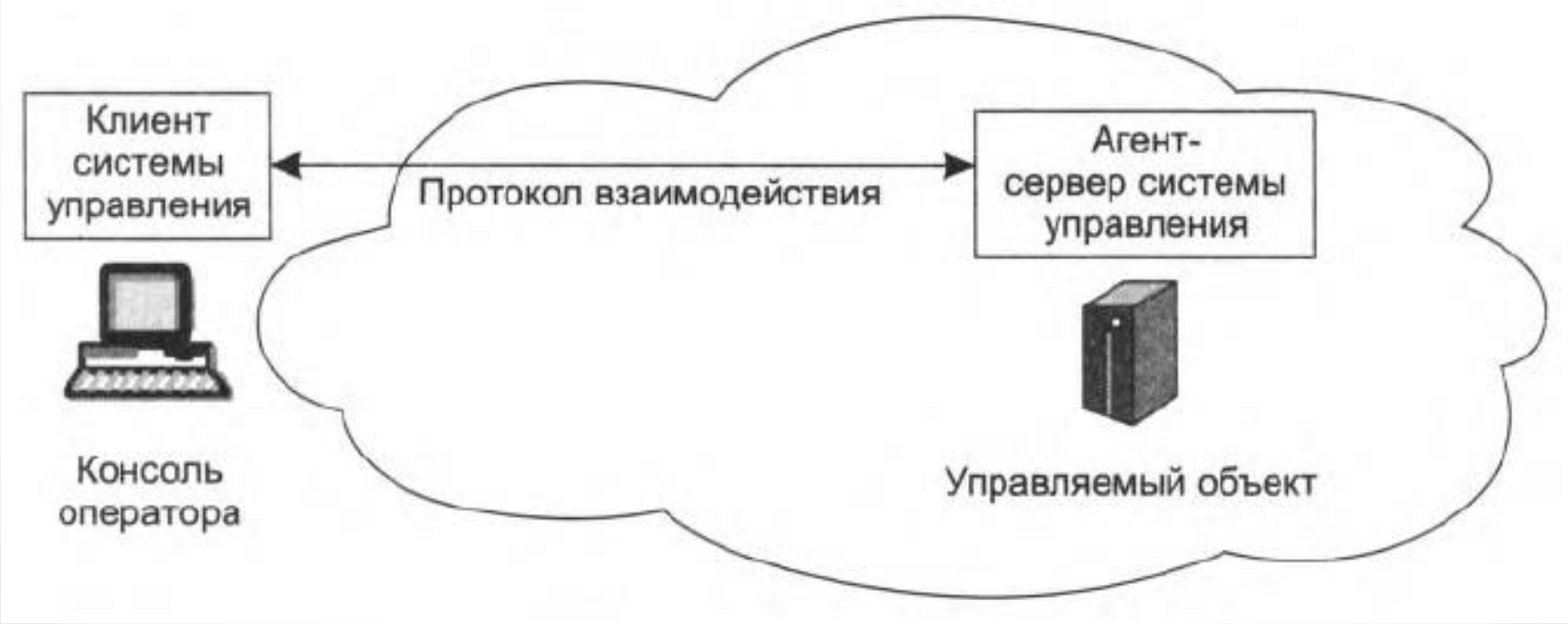
Среди задач, определенных для систем управления сетью, есть сравнительно редкие операции (например, конфигурирование того или иного устройства), а есть и такие, которые требуют частого вмешательства системы (анализ производительности каждого из устройств сети, сбор статистики по загрузке устройств).

В первом случае используется «ручное» управление – администратор со своей консоли передает команды агенту.

Такой вариант называется **ручным двухзвенным управлением**.

Служба управления сетью

Ручное двухзвенное управление устройством



Служба управления сетью

В качестве протокола взаимодействия клиента и сервера может применяться, например, протокол удаленного управления telnet, клиентская часть которого должна быть установлена на компьютере администратора, а серверная – на устройстве.

Серверная часть telnet должна также поддерживать интерфейс с агентом, от которого будет поступать информация о состоянии управляемого объекта и значении его характеристик.

На клиентской стороне протокол telnet может быть связан с программой поддержки графического пользовательского интерфейса.

Служба управления сетью

Для задач, требующих частого выполнения операций управления отдельными устройствами, а также при росте количества управляемых устройств рассмотренная схема уже не может решить поставленную задачу.

В схему вводится новое промежуточное звено – менеджер, призванный автоматизировать взаимодействие оператора с множеством агентов.

Такая схема службы управления сетью реализуется в виде **трехзвенного распределенного приложения**.

Служба управления сетью

Функции между звеньями **трехзвенного распределенного приложения** распределены следующим образом:

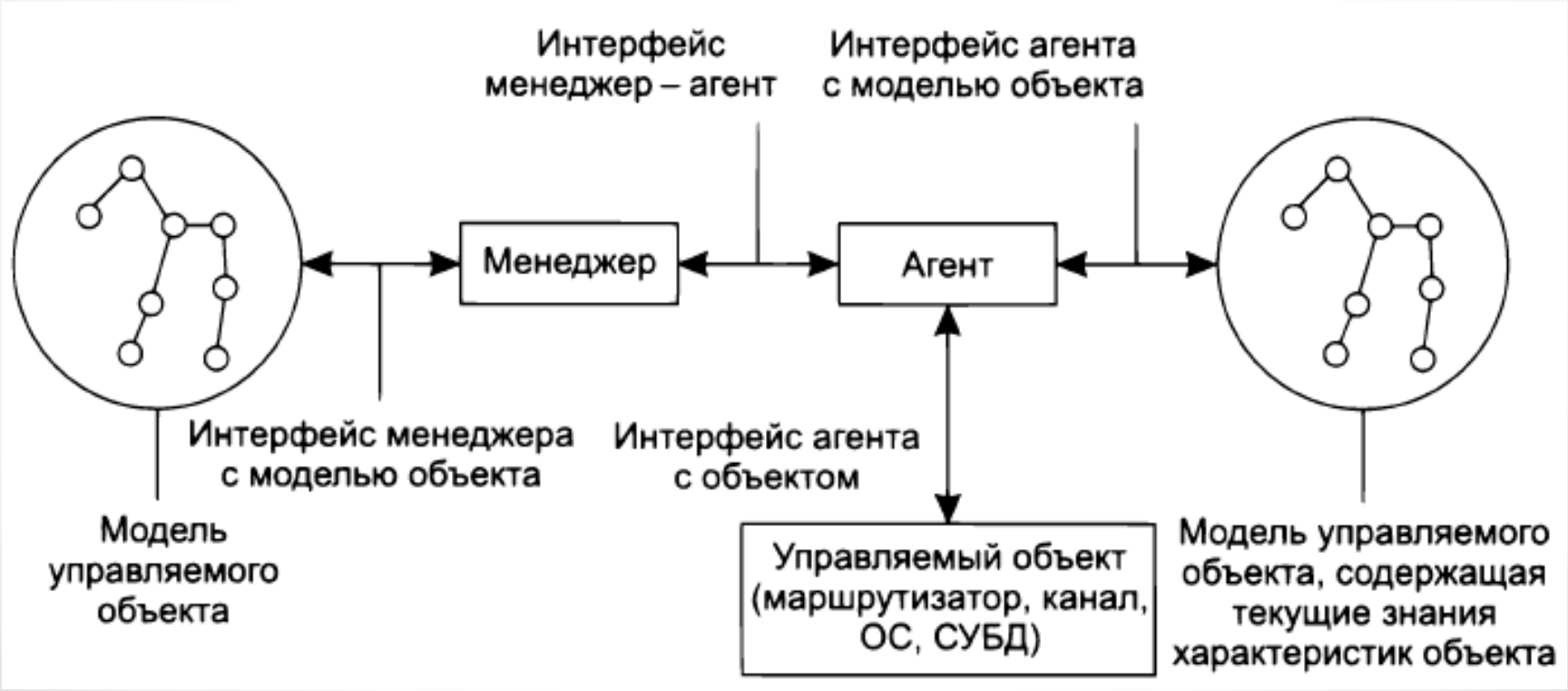
- Первое звено – **клиент системы управления**, устанавливается на компьютере оператора, поддерживает пользовательский интерфейс с промежуточным сервером.
- Второе звено – **промежуточный сервер**, который выполняет функции **менеджера** и устанавливается либо на компьютере оператора, либо на специально выделенном компьютере.
- Третье звено, **агент**, устанавливается на управляемом объекте или связанном с ним компьютере.

Служба управления сетью

Трехзвенная схема управления сетью

Служба управления сетью

Взаимодействие агента, менеджера и управляемого объекта



Служба управления сетью

Для каждого управляемого объекта в сети создается некоторая **модель объекта**, представляющая все характеристики объекта, необходимые для его контроля.

Например, модель маршрутизатора обычно включает такие характеристики, как количество портов, их тип, таблицу маршрутизации, количество кадров и пакетов протоколов канального, сетевого и транспортного уровней, прошедших через эти порты.

Модели объектов сети используются менеджером как источник знаний о том, какой набор характеристик имеет тот или иной объект.

Служба управления сетью

Модель объекта совпадает с логической схемой базы данных (БД) объекта, хранящей значения его характеристик.

Эта БД хранится на устройстве и постоянно пополняется результатами измерений характеристик, которые проводит агент.

В системах управления сетями, построенных на основе протокола SNMP, такая база называется **базой данных управляющей информации** (**Management Information Base**, **MIB).**

Служба управления сетью

Менеджер не имеет непосредственного доступа к базе данных MIB – для получения конкретных значений характеристик объекта он должен по сети обратиться к его агенту.

Агент является посредником между управляемым объектом и менеджером.

Менеджер и агент взаимодействуют по стандартному протоколу, который позволяет менеджеру запрашивать значения параметров, хранящихся в MIB, а агенту – передавать информацию, на основе которой менеджер должен управлять объектом.

Служба управления сетью

Различают **внутриполосное управление** (**In-band**), когда команды управления идут по тому же каналу, по которому передаются пользовательские данные, и **внеполосное управление** (**Out-band**), осуществляемое вне канала передачи пользовательских данных.

Внутриполосное управление более экономично, так как не требует создания отдельной инфраструктуры передачи управляющих данных.

Однако внеполосное управление надежнее, так как соответствующее оборудование может выполнять свои функции даже тогда, когда те или иные сетевые элементы выходят из строя.

Служба управления сетью

Как правило, используются два типа связей между менеджерами:

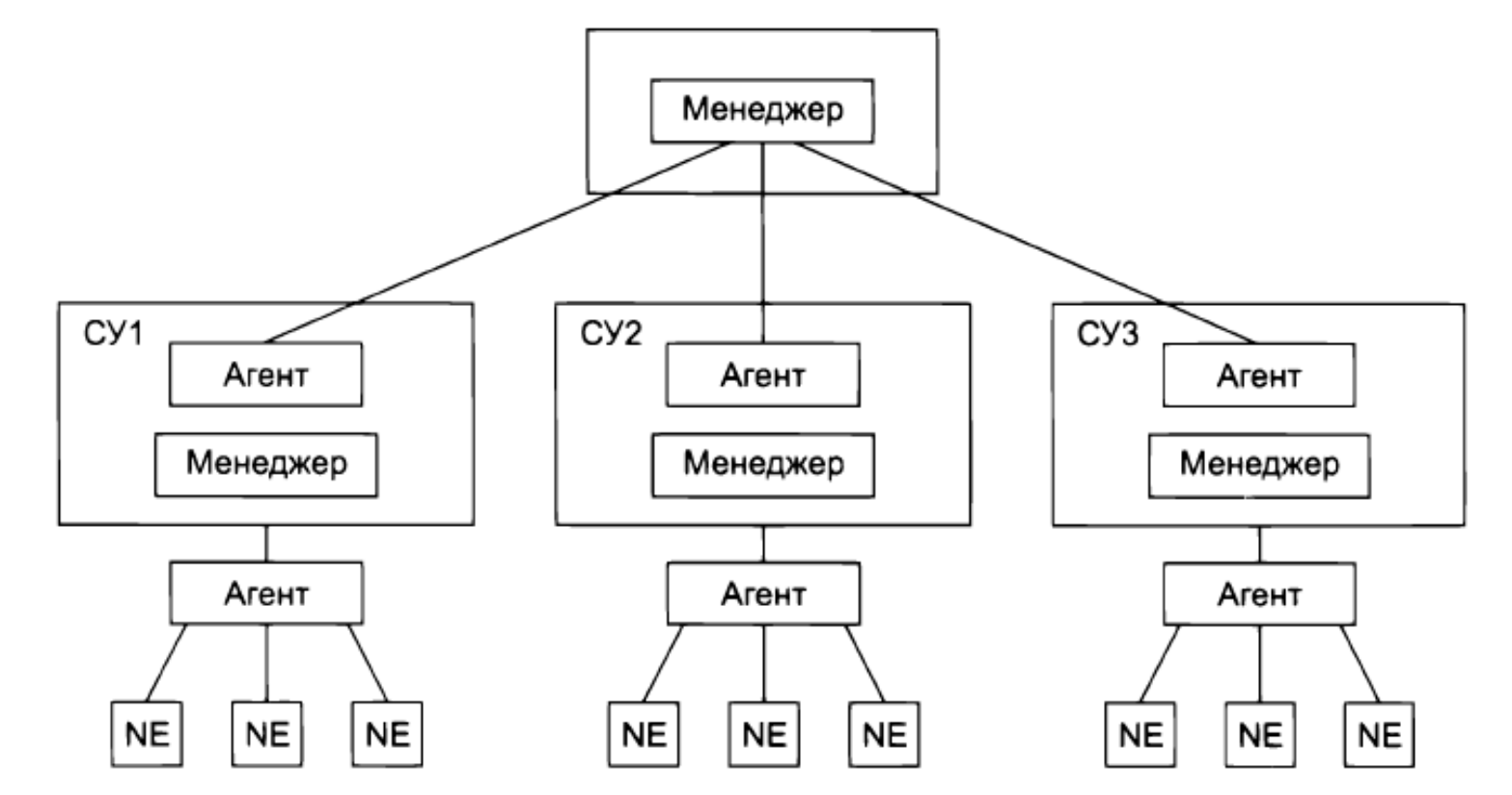
- одноранговая;
- иерархическая.

Служба управления сетью

Одноранговые связи между менеджерами

Служба управления сетью

Иерархические связи между менеджерами



Протокол SNMP

Протокол **SNMP** (**Simple Management Network Protocol – простой протокол сетевого администрирования**) используется в качестве стандартного протокола взаимодействия менеджера и агента.

Протокол SNMP относится к прикладному уровню стека TCP/IP.

Для транспортировки своих сообщений он использует дейтаграммный транспортный протокол UDP.

SNMP – это протокол типа «запрос-ответ», то есть на каждый запрос, поступивший от менеджера, агент должен передать ответ.

Протокол SNMP

Особенностью протокола является его чрезвычайная простота – он включает в себя всего несколько команд:

- Команда **Get Request** используется менеджером для запроса агента о значении какой-либо переменной по ее стандартному имени.
- Команда **GetNextRequest** применяется менеджером для извлечения значения следующего объекта (без указания его имени) при последовательном просмотре таблицы объектов.
- С помощью команды **Response** SNMP-агент передает менеджеру ответ на команду GetRequest или GetNextRequest.

Протокол SNMP

- Команда **SetRequest** позволяет менеджеру изменять значения какой-либо переменной или списка переменных.

С помощью команды SetRequest и происходит собственно управление устройством. Агент должен «понимать» смысл значений переменной, которая используется для управления устройством, и на основании этих значений выполнять реальное управляющее воздействие.

- Команда **Trap** используется агентом для сообщения менеджеру о возникновении особой ситуации.
- Команда **GetBulk** позволяет менеджеру получить несколько переменных за один запрос.

Протокол SNMP

Любое SNMP-сообщение состоит из трех основных частей:

- **версии протокола**;
- **общей строки**;
- **области данных**.

Протокол SNMP

Общая строка (**community string**) используется для группирования устройств, управляемых определенным менеджером.

Общая строка является своего рода паролем, так как для того, чтобы устройства могли взаимодействовать по протоколу SNMP, они должны иметь одно и то же значение этого идентификатора (по умолчанию часто употребляется строка «public»).

Протокол SNMP

В **области данных** содержатся описанные команды протокола, а также имена объектов и их значения.

Область данных состоит из одного или более блоков, каждый из которых может относиться к одному из перечисленных типов команд протокола SNMP.

Для каждого типа команды определен свой формат.

Протокол telnet

Режим удаленного управления, называемый также режимом **терминального доступа,** предполагает, что пользователь превращает свой компьютер в виртуальный терминал другого компьютера, к которому он получает удаленный доступ.

Режим удаленного управления реализуется специальным протоколом прикладного уровня, работающим поверх транспортных протоколов, которые связывают удаленный узел с компьютерной сетью.

Для IP-сетей наиболее старым протоколом этого типа является telnet (RFC 854).

Протокол telnet

Протокол **telnet** работает в архитектуре клиент-сервер, обеспечивая эмуляцию алфавитно-цифрового терминала и ограничивая пользователя режимом командной строки.

При нажатии клавиши соответствующий код перехватывается клиентом telnet, помещается в TCP-сообщение и отправляется через сеть узлу, которым пользователь хочет управлять.

При поступлении на узел назначения код нажатой клавиши извлекается из TCP-сообщения сервером telnet и передается ОС узла, которая рассматривает сеанс telnet как один из сеансов локального пользователя.

Протокол telnet

Если ОС реагирует на нажатие клавиши выводом очередного символа на экран, то для сеанса удаленного пользователя этот символ также упаковывается в TCP-сообщение и по сети отправляется удаленному узлу.

Клиент telnet извлекает символ и отображает его в окне своего терминала, эмулируя терминал удаленного узла.

Протокол telnet

Протокол telnet был реализован в среде **Unix** и, наряду с электронной почтой и FTP-доступам к архивам файлов, был популярным сервисом Интернета.

Однако, поскольку для аутентификации пользователей в технологии telnet применяются пароли, передаваемые через сеть в виде обычного текста, telnet сейчас работает преимущественно в пределах одной локальной сети, где возможностей для перехвата пароля гораздо меньше.

Протокол SSH

Для удаленного управления узлами через Интернет вместо telnet обычно применяется протокол **SSH** (**Secure SHell**), который, как и telnet, был первоначально разработан для ОС Unix.

SSH, как и telnet, передает набираемые на терминале пользователя символы на удаленный узел без интерпретации их содержания.

Однако в SSH предусмотрены меры по защите передаваемых аутентификационных и пользовательских данных.

Протокол telnet

Сегодня основной областью применения telnet является управление не компьютерами, а коммуникационными устройствами: маршрутизаторами, коммутаторами и хабами.

Таким образом, он уже скорее представляет собой не пользовательский протокол, а протокол администрирования, то есть альтернативу SNMP.

Протокол telnet

Тем не менее отличие между протоколами telnet и SNMP принципиально!

Telnet предусматривает обязательное участие человека в процессе администрирования, так как, по сути, лишь транслирует команды, которые вводит администратор при конфигурировании или мониторинге маршрутизатора (иного коммуникационного устройства).

Протокол SNMP, наоборот, рассчитан на автоматические процедуры мониторинга и управления, хотя и не исключает возможности участия администратора в этом процессе.