Вопрос 3   
Многоуровневый подход к организации сетевого взаимодействия. Модель ISO/OSI. Протоколы, интерфейсы  
("Общие принципы построения вычислительных сетей")

**1.3. Понятие «открытая система» и проблемы стандартизации**

* [**1.3.1. Многоуровневый подход. Протокол. Интерфейс. Стек протоколов**](http://compnets.narod.ru/1-6.html#1.3.1.%20%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9%20%D0%BF%D0%BE%D0%B4%D1%85%D0%BE%D0%B4.%20%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB.%20%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81.%20%D0%A1%D1)
* [**1.3.2. Модель OSI**](http://compnets.narod.ru/1-6.html#1.3.2.%20%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C%20OSI)

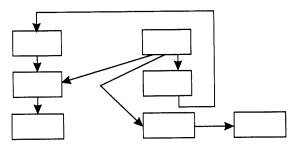
**1.3. Понятие «открытая система» и проблемы стандартизации**

Универсальный тезис о пользе стандартизации, справедливый для всех отраслей, в компьютерных сетях приобретает особое значение. Суть сети - это соединение разного оборудования, а значит, проблема совместимости является одной из наиболее острых. Без принятия всеми производителями общепринятых правил построения оборудования прогресс в деле «строительства» сетей был бы невозможен. Поэтому все развитие компьютерной отрасли в конечном счете отражено в стандартах - любая новая технология только тогда приобретает «законный» статус, когда ее содержание закрепляется в соответствующем стандарте.

В компьютерных сетях *идеологической основой* стандартизации является **многоуровневый подход к разработке средств сетевого взаимодействия**. Именно на основе этого подхода была разработана стандартная **семиуровневая модель** взаимодействия открытых систем, ставшая своего рода универсальным языком сетевых специалистов.

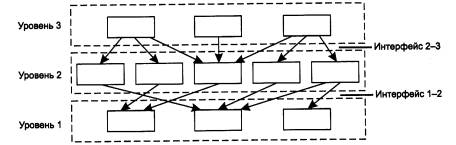
**1.3.1. Многоуровневый подход. Протокол. Интерфейс. Стек протоколов**

Организация взаимодействия между устройствами в сети является сложной задачей. Как известно, для решения сложных задач используется универсальный прием - декомпозиция, то есть разбиение одной сложной задачи на несколько более простых задач-модулей (рис. 1.20). Процедура декомпозиции включает в себя четкое определение функций каждого модуля, решающего отдельную задачу, и интерфейсов между ними. В результате достигается логическое упрощение задачи, а кроме того, появляется возможность модификации отдельных модулей без изменения остальной части системы.



**Рис. 1.20.** Пример декомпозиции задачи

При декомпозиции часто используют многоуровневый подход. Он заключается в следующем. Все множество модулей разбивают на уровни. Уровни образуют иерархию, то есть имеются вышележащие и нижележащие уровни (рис. 1.21). Множество модулей, составляющих каждый уровень, сформировано таким образом, что для выполнения своих задач они обращаются с запросами только к модулям непосредственно примыкающего нижележащего уровня. С другой стороны, результаты работы всех модулей, принадлежащих некоторому уровню, могут быть переданы только модулям соседнего вышележащего уровня. Такая иерархическая декомпозиция задачи предполагает четкое определение функции каждого уровня и интерфейсов между уровнями. Интерфейс определяет набор функций, которые нижележащий уровень предоставляет вышележащему. В результате иерархической декомпозиции достигается относительная независимость уровней, а значит, и возможность их легкой замены.



**Рис. 1.21.** Многоуровневый подход - создание иерархии задач

Средства сетевого взаимодействия, конечно, тоже могут быть представлены в виде иерархически организованного множества модулей.

При этом *модули нижнего уровня* могут, например, **решать все вопросы, связанные с надежной передачей электрических сигналов между двумя соседними узлами**.

Модули более *высокого уровня* **организуют транспортировку сообщений в пределах всей сети**, пользуясь для этого средствами упомянутого нижележащего уровня.

А *на верхнем уровне* **работают модули, предоставляющие пользователям доступ к различным службам - файловой, печати и т. п.**

Конечно, это только один из множества возможных вариантов деления общей задачи организации сетевого взаимодействия на частные подзадачи.

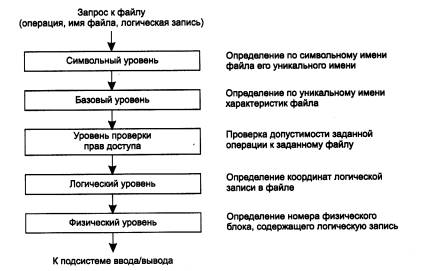
**Многоуровневый** подход к описанию и реализации функций системы применяется не только в отношении сетевых средств.

Такая модель функционирования используется, например, в локальных файловых системах, когда поступивший запрос на доступ к файлу последовательно обрабатывается несколькими программными уровнями (рис. 1.22).

1. Запрос вначале анализируется верхним уровнем, на котором осуществляется последовательный разбор составного символьного имени файла и определение уникального идентификатора файла.

2. Следующий уровень находит по уникальному имени все основные характеристики файла: адрес, атрибуты доступа и т. п.

3. Затем на более низком уровне осуществляется проверка прав доступа к этому файлу, а далее, после расчета координат области файла, содержащей требуемые данные, выполняется физический обмен с внешним устройством с помощью драйвера диска.

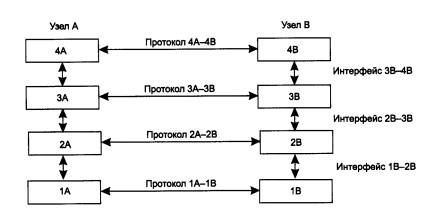


**Рис. 1.22.** Многоуровневая модель файловой системы

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют **две машины,** то есть в данном случае необходимо организовать согласованную работу двух «иерархий». При передаче сообщений оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения длины сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты для всех уровней, начиная от самого низкого - уровня передачи битов - до самого высокого, реализующего сервис для пользователей сети.

На рис. 1.23 показана модель взаимодействия двух узлов. С каждой стороны средства взаимодействия представлены четырьмя уровнями.

Процедура взаимодействия этих двух узлов может быть описана в виде набора правил взаимодействия каждой пары соответствующих уровней обеих участвующих сторон. Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, **называются *протоколом*.**



**Рис. 1.23.** Взаимодействие двух узлов

Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другом в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений. **Эти правила принято называть *интерфейсом*.**

Интерфейс определяет набор сервисов, предоставляемый данным уровнем соседнему уровню. В сущности, протокол и интерфейс выражают одно и то же понятие, но традиционно в сетях за ними закрепили разные области действия: протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы - модулей соседних уровней в одном узле.

Средства каждого уровня должны отрабатывать, во-первых, свой собственный протокол, а во-вторых, интерфейсы с соседними уровнями.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, **называется *стеком коммуникационных протоколов*.**

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно.

Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней - как правило, чисто программными средствами.

Программный модуль, реализующий некоторый протокол, часто для краткости также называют **«протоколом».** При этом соотношение между протоколом - формально определенной процедурой и протоколом - программным модулем, реализующим эту процедуру, аналогично соотношению между алгоритмом решения некоторой задачи и программой, решающей эту задачу.

Понятно, что один и тот же алгоритм может быть запрограммирован с разной степенью эффективности. Точно так же и протокол может иметь несколько программных реализации. Именно поэтому при сравнении протоколов следует учитывать не только логику их работы, но и качество программных решений. Более того, на эффективность взаимодействия устройств в сети влияет качество всей совокупности протоколов, составляющих стек, в частности, насколько рационально распределены функции между протоколами разных уровней и насколько хорошо определены интерфейсы между ними.

Протоколы реализуются не только компьютерами, но и другими сетевыми устройствами - концентраторами, мостами, коммутаторами, маршрутизаторами и т. д. Действительно, в общем случае связь компьютеров в сети осуществляется не напрямую, а через различные коммуникационные устройства. В зависимости от типа устройства в нем должны быть встроенные средства, реализующие тот или иной набор протоколов.

Чтобы еще раз пояснить понятия «протокол» и «интерфейс», рассмотрим пример, не имеющий отношения к вычислительным сетям, а именно обсудим взаимодействие двух предприятий А и В; связанных между собой деловым сотрудничеством. Между предприятиями существуют многочисленные договоренности и соглашения, такие, например, как регулярные поставки продукции одного предприятия другому. В соответствии с этой договоренностью начальник отдела продаж предприятия А регулярно в начале каждого месяца посылает официальное сообщение начальнику отдела закупок предприятия В о том, сколько и какого товара может быть поставлено в этом месяце. В ответ на это сообщение начальник отдела закупок предприятия В посылает в ответ заявку установленного образца на требуемое количество продукции. Возможно, процедура взаимодействия этих начальников включает дополнительные согласования, в любом случае существует установленный порядок взаимодействия, который можно считать «протоколом уровня начальников». Начальники посылают свои сообщения и заявки через своих секретарей. Порядок взаимодействия начальника и секретаря соответствует понятию межуровневого интерфейса «начальник - секретарь». На предприятии А обмен документами между начальником и секретарем идет через специальную папку, а на предприятии В начальник общается с секретарем по факсу. Таким образом, интерфейсы «начальник - секретарь» на этих двух предприятиях отличаются.

После того как сообщения переданы секретарям, начальников не волнует, каким образом эти сообщения будут перемещаться дальше - обычной или электронной почтой, факсом или нарочным. Выбор способа передачи - это уровень компетенции секретарей, они могут решать этот вопрос, не уведомляя об этом своих начальников, так как их протокол взаимодействия связан только с передачей сообщений, поступающих сверху, и не касается содержания этих сообщений. На рис. 1.24 показано, что в качестве протокола взаимодействия «секретарь-секретарь» используется обмен письмами. При решении других вопросов начальники могут взаимодействовать по другим правилам-протоколам, но это не повлияет на работу секретарей, для которых не важно, какие сообщения отправлять, а важно, чтобы они дошли до адресата. Итак, в данном случае мы имеем дело с двумя уровнями - уровнем начальников и уровнем секретарей, и каждый из них имеет собственный протокол, который может быть изменен независимо от протокола другого уровня. Эта независимость протоколов друг от друга и делает привлекательным многоуровневый подход.



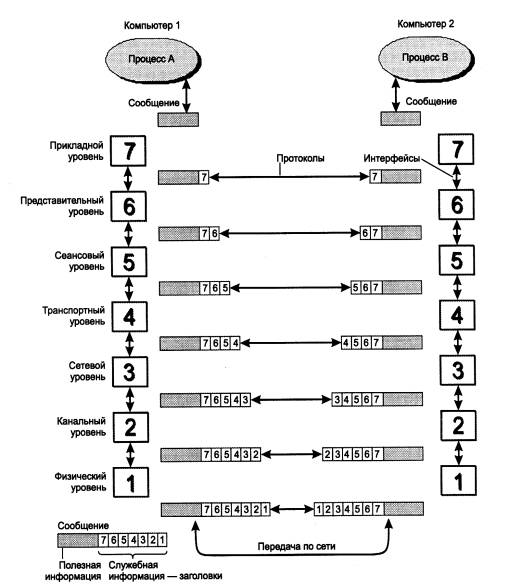
**Рис. 1.24.** Пример многоуровневого взаимодействия предприятий

**1.3.2. Модель OSI**

Из того, что протокол является соглашением, принятым двумя взаимодействующими объектами, в данном случае двумя работающими в сети компьютерами, совсем не следует, что он обязательно является стандартным. Но на практике при реализации сетей стремятся использовать стандартные протоколы. Это могут быть фирменные, национальные или международные стандарты.

В начале 80-х годов ряд международных организаций по стандартизации - ISO, ITU-T и некоторые другие - разработали модель, которая сыграла значительную роль в развитии сетей. Эта модель называется *моделью взаимодействия открытых систем (Open System Interconnection, OSI)* или моделью OSI. Модель OSI определяет различные уровни взаимодействия систем, дает им стандартные имена и указывает, какие функции должен выполнять каждый уровень. Модель OSI была разработана на основании большого опыта, полученного при создании компьютерных сетей, в основном глобальных, в 70-е годы. Полное описание этой модели занимает более 1000 страниц текста.

В модели OSI (рис. 1.25) средства взаимодействия делятся на семь уровней: прикладной, представительный, сеансовый, транспортный, сетевой, канальный и физический. Каждый уровень имеет дело с одним определенным аспектом взаимодействия сетевых устройств.



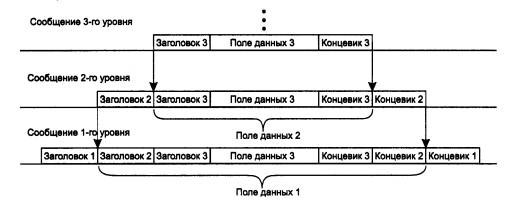
**Рис. 1.25.** Модель взаимодействия открытых систем ISO/OSI

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей. Свои собственные протоколы взаимодействия приложения реализуют, обращаясь к системным средствам. Поэтому необходимо различать уровень взаимодействия приложений и прикладной уровень.

Следует также иметь в виду, что приложение может взять на себя функции некоторых верхних уровней модели OSI. Например, некоторые СУБД имеют встроенные средства удаленного доступа к файлам. В этом случае приложение, выполняя доступ к удаленным ресурсам, не использует системную файловую службу; оно обходит верхние уровни модели OSI и обращается напрямую к системным средствам, ответственным за транспортировку сообщений по сети, которые располагаются на нижних уровнях модели OSI.

Итак, пусть приложение обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. Обычное сообщение состоит из заголовка и поля данных. Заголовок содержит служебную информацию, которую необходимо передать через сеть прикладному уровню машины-адресата, чтобы сообщить ему, какую работу надо выполнить. В нашем случае заголовок, очевидно, должен содержать информацию о месте нахождения файла и о типе операции, которую необходимо над ним выполнить. Поле данных сообщения может быть пустым или содержать какие-либо данные, например те, которые необходимо записать в удаленный файл. Но для того чтобы доставить эту информацию по назначению, предстоит решить еще много задач, ответственность за которые несут нижележащие уровни.

После формирования сообщения прикладной уровень направляет его вниз по стеку представительному уровню. Протокол представительного уровня на основании информации, полученной из заголовка прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию - заголовок представительного уровня, в котором содержатся указания для протокола представительного уровня машины-адресата. Полученное в результате сообщение передается вниз сеансовому уровню, который в свою очередь добавляет свой заголовок, и т. д. (Некоторые реализации протоколов помещают служебную информацию не только в начале сообщения в виде заголовка, но и в конце, в виде так называемого «концевика».) Наконец, сообщение достигает нижнего, физического уровня, который собственно и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рис. 1.26).



**Рис. 1.26.** Вложенность сообщений различных уровней

Когда сообщение по сети поступает на машину - адресат, оно принимается ее физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие данному уровню функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню.

Наряду с термином *сообщение (message)* существуют и другие термины, применяемые сетевыми специалистами для обозначения единиц данных в процедурах обмена. В стандартах ISO для обозначения единиц данных, с которыми имеют дело протоколы разных уровней, используется общее название *протокольный блок данных (Protocol Data Unit, PDU)*. Для обозначения блоков данных определенных уровней-часто используются специальные названия: кадр (frame), пакет (packet), дейтаграмма (datagram), сегмент (segment).

В модели OSI различаются два основных типа протоколов. В протоколах с *установлением соединения (connection-oriented)* перед обменом данными отправитель и получатель должны сначала установить соединение и, возможно, выбрать некоторые параметры протокола, которые они будут использовать при обмене данными. После завершения диалога они должны разорвать это соединение. Телефон - это пример взаимодействия, основанного на установлении соединения.

Вторая группа протоколов - протоколы *без предварительного установления соединения (connectionless)*. Такие протоколы называются также *дейтаграммными* протоколами. Отправитель просто передает сообщение, когда оно готово. Опускание письма в почтовый ящик - это пример связи без предварительного установления соединения. При взаимодействии компьютеров используются протоколы обоих типов.

# [Модель OSI. 7 уровней сетевой модели OSI с примерами (selectel.ru)](https://selectel.ru/blog/osi-for-beginners/) Простое пособие по сетевой модели OSI для начинающих