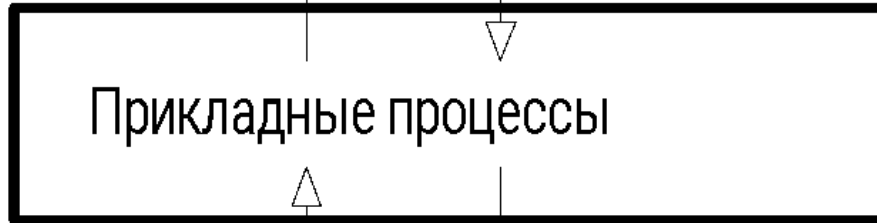


Семиуровневая модель OSI

- Для единого представления данных в сетях с неоднородными устройствами и программным обеспечением международная организация по стандартам ISO (International Standardization Organization) разработала базовую модель связи открытых систем OSI (Open System Interconnection). Эта модель описывает правила и процедуры передачи данных в различных сетевых средах при организации сеанса связи. Основными элементами модели являются уровни, прикладные процессы и физические средства соединения. На рис. 2.1 представлена структура базовой модели. Каждый уровень модели OSI выполняет определенную задачу в процессе передачи данных по сети. Базовая модель является основой для разработки сетевых протоколов. OSI разделяет коммуникационные функции в сети на семь уровней, каждый из которых обслуживает различные части процесса области взаимодействия открытых систем.

Пользователи

**Интерфейс
пользователя**



**Прикладной
интерфейс**

Уровни:

7

Прикладной

6

Представительский

5

Сеансовый

4

Транспортный

3

Сетевой

2

Канальный

1

Физический

**Область
взаимодействия
открытых
систем**

Физические средства соединения

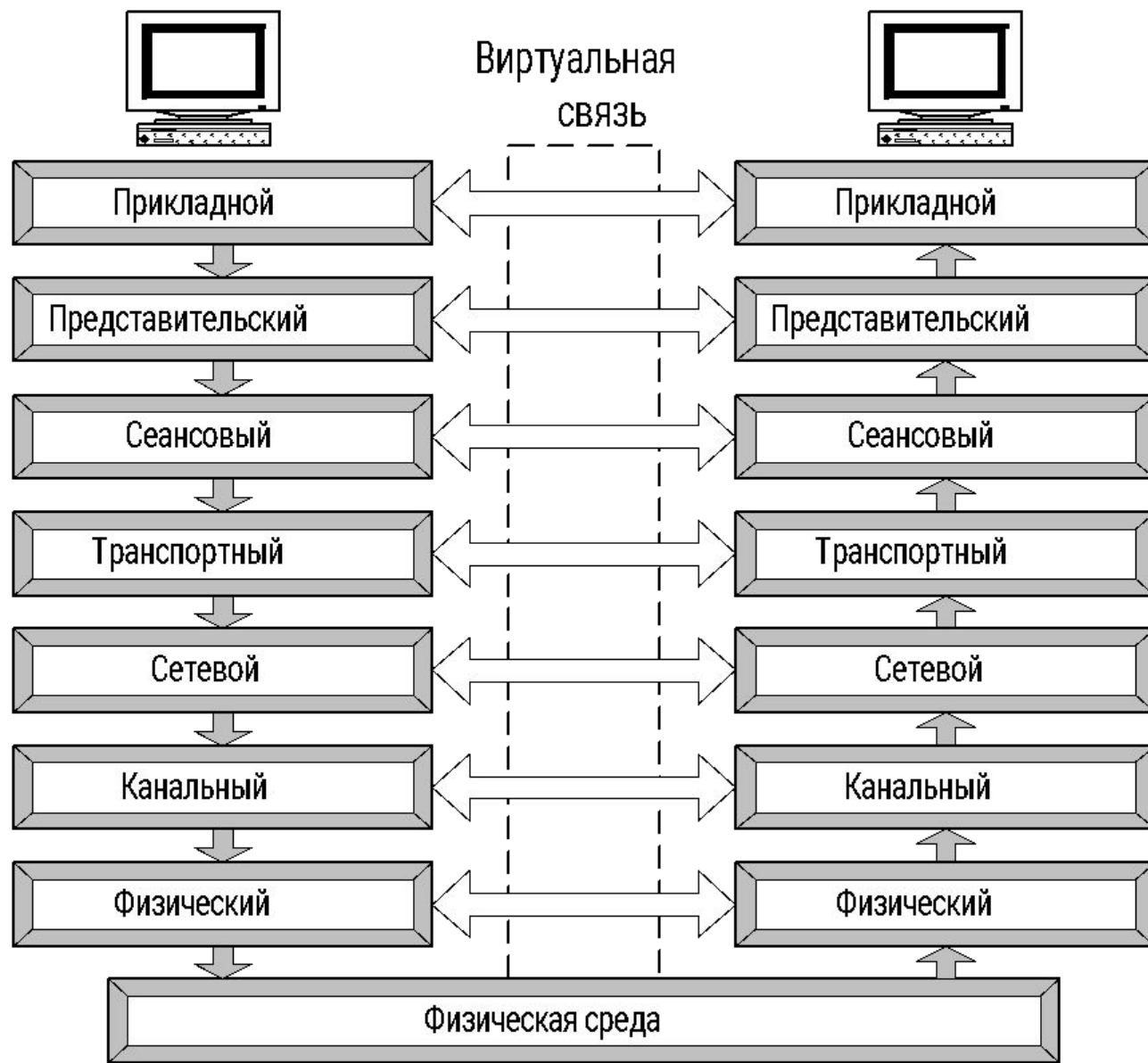
- Модель OSI описывает только системные средства взаимодействия, не касаясь приложений конечных пользователей. Приложения реализуют свои собственные протоколы взаимодействия, обращаясь к системным средствам. Если приложение может взять на себя функции некоторых верхних уровней модели OSI, то для обмена данными оно обращается напрямую к системным средствам, выполняющим функции оставшихся нижних уровней модели OSI.

Взаимодействие уровней модели OSI

- Модель OSI можно разделить на две различных модели, как показано на рис.2.2:
- горизонтальную модель на базе протоколов, обеспечивающую механизм взаимодействия программ и процессов на различных машинах;
- вертикальную модель на основе услуг, обеспечиваемых соседними уровнями друг другу на одной машине.

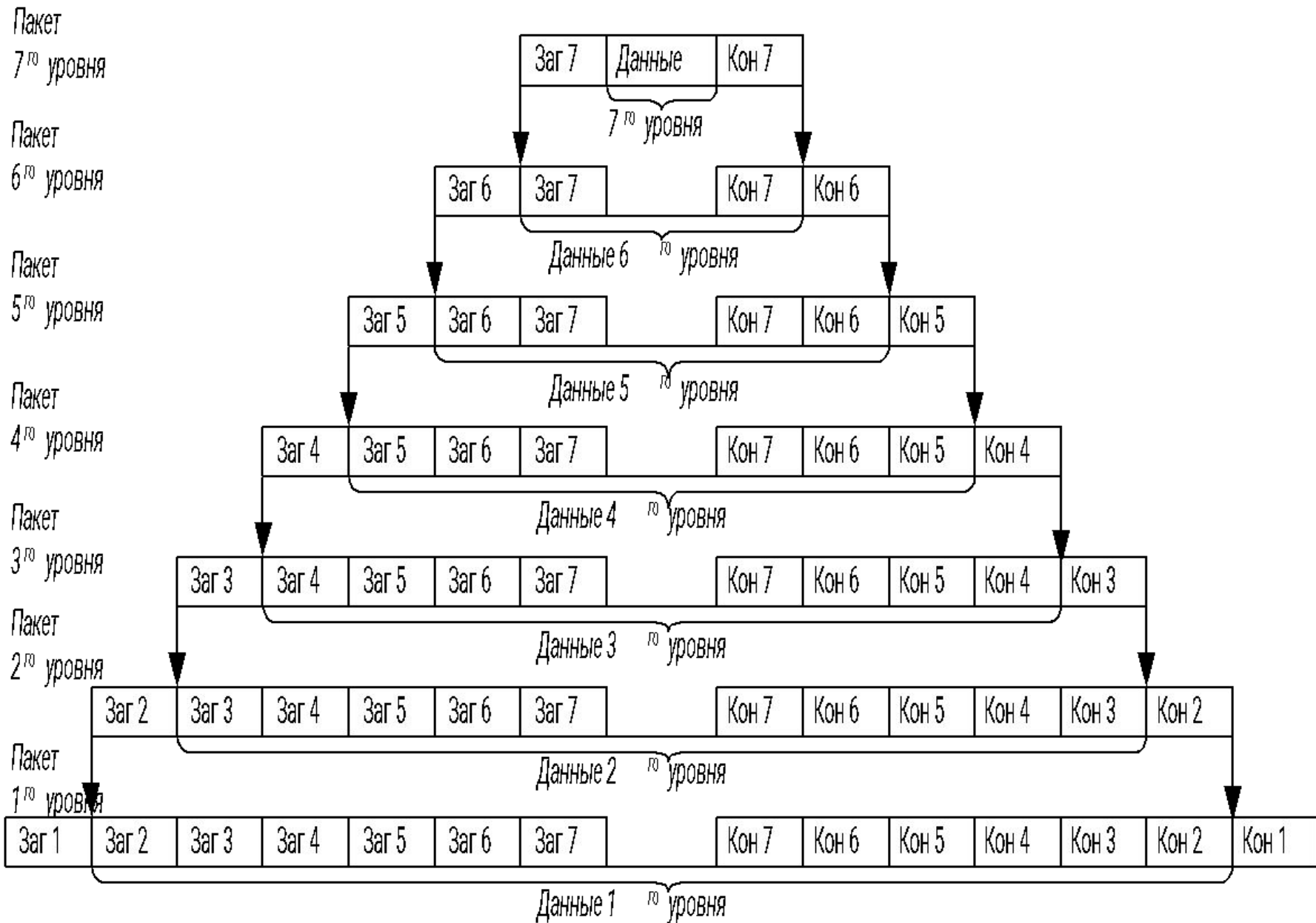
Компьютер-отправитель

Компьютер-получатель



- Каждый уровень компьютера–отправителя взаимодействует с таким же уровнем компьютера-получателя, как будто он связан напрямую. Такая связь называется логической или виртуальной связью. В действительности взаимодействие осуществляется между смежными уровнями одного компьютера.
- В горизонтальной модели двум программам требуется общий протокол для обмена данными. В вертикальной модели соседние уровни обмениваются данными с использованием интерфейсов прикладных программ API (Application Programming Interface).

- Перед подачей в сеть данные разбиваются на пакеты. Пакет (packet) – это единица информации, передаваемая между станциями сети. При отправке данных пакет проходит последовательно через все уровни программного обеспечения. На каждом уровне к пакету добавляется управляющая информация данного уровня (заголовок), которая необходима для успешной передачи данных по сети, как это показано на рис. 2.3, где *Заг* – заголовок пакета, *Кон* – конец пакета.



- Каждый уровень модели выполняет свою функцию. Чем выше уровень, тем более сложную задачу он решает.
- Отдельные уровни модели *OSI* удобно рассматривать как *группы программ*, предназначенных для выполнения конкретных *функций*. Один уровень, к примеру, отвечает за обеспечение преобразования данных из *ASCII* в *EBCDIC* и содержит *программы* необходимые для выполнения этой задачи.
- Каждый уровень обеспечивает сервис для вышестоящего уровня, запрашивая в свою очередь, сервис у нижестоящего уровня. Верхние уровни запрашивают сервис почти одинаково: как правило, это требование маршрутизации каких-то данных из одной сети в другую. Практическая реализация принципов адресации данных возложена на нижние уровни.

- Рассматриваемая модель определяет взаимодействие открытых систем разных производителей в одной сети. Поэтому она выполняет для них координирующие действия по:
 - взаимодействию прикладных процессов;
 - формам представления данных;
 - единообразному хранению данных;
 - управлению сетевыми ресурсами;
 - безопасности данных и защите информации;
 - диагностике программ и технических средств.

приведено краткое описание функций всех уровней

7. Прикладной

представляет набор интерфейсов, позволяющий получить доступ к сетевым службам

6. Представления

преобразует данные в общий формат для передачи по сети

5. Сеансовый

поддержка взаимодействия (сеанса) между удаленными процессами

4. Транспортный

управляет передачей данных по сети, обеспечивает подтверждение передачи

3. Сетевой

маршрутизация, управление потоками данных, адресация сообщений для доставки, преобразование логические сетевые адреса и имена в соответствующие им физические

2. Канальный

2.1. Контроль логической связи (LLC):

формирование кадров

2.2. Контроль доступа к среде (MAC):

управление доступом к среде

1. Физический:

битовые протоколы передачи информации

Прикладной уровень (Application layer)

Прикладной уровень обеспечивает прикладным процессам средства доступа к области взаимодействия, является верхним (седьмым) уровнем и непосредственно примыкает к прикладным процессам. В действительности прикладной уровень – это набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например с помощью протокола электронной почты [30]. Специальные элементы прикладного сервиса обеспечивают сервис для конкретных прикладных программ, таких как программы пересылки файлов и эмуляции терминалов. Если, например программе необходимо переслать файлы, то обязательно будет использован *протокол передачи, доступа и управления файлами FTAM (File Transfer, Access, and Management)*. В модели OSI *прикладная программа*, которой нужно выполнить конкретную задачу (например, обновить базу данных на компьютере), посылает конкретные данные в виде *Дейтаграммы на прикладной уровень*. Одна из основных задач этого уровня - определить, как следует обрабатывать запрос прикладной программы, другими словами, какой вид должен принять данный запрос.

Единица данных, которой оперирует прикладной уровень, обычно называется сообщением (message).

Описание форм и методов взаимодействия прикладных процессов

- Выполнение различных видов работ.
- передача файлов;
- управление заданиями;
- управление системой и т.д.
- Идентификация пользователей по их паролям, адресам, электронным подписям;
- Определение функционирующих абонентов и возможности доступа к новым прикладным процессам;
- Определение достаточности имеющихся ресурсов;
- Организация запросов на соединение с другими прикладными процессами;
- Передача заявок представительскому уровню на необходимые методы описания информации;
- Выбор процедур планируемого диалога процессов;
- Управление данными, которыми обмениваются прикладные процессы и синхронизация взаимодействия прикладных процессов;
- Определение качества обслуживания (время доставки блоков данных, допустимой частоты ошибок);
- Соглашение об исправлении ошибок и определении достоверности данных;
- Согласование ограничений, накладываемых на синтаксис (наборы символов, структура данных).

- На *прикладном уровне* необходимо предоставить в распоряжение пользователей уже переработанную информацию. С этим может справиться системное и пользовательское программное обеспечение.
- Прикладной уровень отвечает за доступ приложений в сеть. Задачами этого уровня является перенос файлов, обмен почтовыми сообщениями и управление сетью.

- К числу наиболее распространенных протоколов верхних трех уровней относятся:
- FTP (File Transfer Protocol) протокол передачи файлов;
- TFTP (Trivial File Transfer Protocol) простейший протокол пересылки файлов;
- X.400 электронная почта;
- Telnet работа с удаленным терминалом;
- SMTP (Simple Mail Transfer Protocol) простой протокол почтового обмена;
- CMIP (Common Management Information Protocol) общий протокол управления информацией;
- SLIP (Serial Line IP) IP для последовательных линий. Протокол последовательной посимвольной передачи данных;
- SNMP (Simple Network Management Protocol) простой протокол сетевого управления;
- FTAM (File Transfer, Access, and Management) протокол передачи, доступа и управления файлами.