

Лекция 3. ОСНОВНЫЕ ЗАДАЧИ ПОСТРОЕНИЯ СЕТЕЙ

При создании вычислительных сетей разработчикам пришлось решать множество самых различных задач, связанных с кодированием и синхронизацией электрических (оптических) сигналов, выбором конфигурации физических и логических связей, разработкой схем адресации устройств, созданием различных способов коммутации, мультиплексирования и демультиплексирования потоков данных, совместным использованием передающей среды. В данной лекции мы сформулируем все эти задачи, причем в той последовательности, в которой они возникали в процессе развития и совершенствования сетевых технологий.

Начнем с наиболее простого случая непосредственного соединения двух устройств физическим каналом, такое соединение называется связью "точка-точка" (point-to-point).

Связь компьютера с периферийными устройствами

Частным случаем связи "точка-точка" является соединение компьютера с **периферийным устройством**. Поскольку механизмы взаимодействия компьютеров в сети многое позаимствовали у схемы взаимодействия компьютера с периферийными устройствами, начнем рассматривать принципы работы сети из этого "досетевого" случая.

Для обмена данными компьютер и периферийное устройство (ПП) оснащены внешними **интерфейсами или портами**. В этом случае к понятию "интерфейс" относятся:

- электрический разъем;
- набор проводов, соединяющих устройства;
- совокупность правил обмена информацией из этих проводов.

Со стороны компьютера логике передачи сигналов на внешний интерфейс управляют:

- контроллер ЧП - аппаратный блок, часто реализован в виде отдельной платы;
- драйвер ЧП - программа, управляющая контроллером периферийного устройства.

Со стороны ПУ интерфейс чаще всего реализуется аппаратным устройством управления ПП, хотя встречаются и программно-управляемые периферийные устройства.

Обмен данными между ПП и компьютером, как правило, двунаправленный. Так, например, даже принтер, который представляет собой устройство вывода информации, возвращает в компьютер данные о своем состоянии.

Таким образом, по каналу, связываает внешние интерфейсы, передается следующая информация:

- данные, поступающие от контроллера на ЧП, например байты текста, нужно распечатать на бумаге;
- команды управления, контроллер передает на устройство управления ПП; в ответ на них оно выполняет специальные действия, например переводит головку диска на соответствующую дорожку или выталкивает из принтера лист бумаги;
- данные, возвращаются устройством управления ЧП в ответ на запрос от контроллера, например данные о готовности к выполнению операции.

Рассмотрим последовательность действий, выполняемых в том случае, когда некотором приложении нужно напечатать текст на принтере. Со стороны компьютера в выполнении этой операции участвует, кроме уже названных контроллера, драйвера и приложения, еще один важный компонент - операционная система. Поскольку все операции ввода-вывода являются привилегированными, все приложения при выполнении операций с периферийными устройствами под управлением ОС в качестве арбитра. Итак, последовательность действий такова:

1. Приложение обращается с запросом на выполнение операции печати в операционной системы. В запросе указываются: адрес данных в оперативной памяти информация, идентифицирующая, принтера и операция, которую нужно выполнить.
2. Получив запрос, операционная система анализирует его, решает, может ли он быть выполнен, и если решение положительное, то запускает соответствующий драйвер, передавая ему в качестве параметров адрес выводимых данных. Дальнейшие действия, относящиеся к операции ввода вывода со стороны компьютера реализуются совместно драйвером и контроллером принтера.
3. Драйвер передает команды и данные контроллеру, который помещает их в свой внутренний буфер. Пусть, например, драйвер загружает значение некоторого байта в буфер контроллера ЧП.
4. Контроллер перемещает данные из внутреннего буфера во внешней порт.
5. Контроллер начинает последовательно передавать биты в линию связи, представляя каждый бит соответствующим электрическим сигналом. Чтобы сообщить устройство управления принтера о том, что начинается передача байта, перед передачей первого бита данных контроллер формирует стартовый сигнал специфической формы, а после передачи последнего информационного бита - стоповый сигнал. Эти сигналы синхронизируют передачу байта. Кроме информационных бит, контроллер может передавать бит контроля четности для повышения достоверности обмена.
6. Устройство управления принтера, обнаружив на соответствующей линии стартовый бит, выполняет подготовительные действия и начинает принимать информационные биты, формируя из них байт в своем приемном буфере. Если передача сопровождается битом четности, то выполняется проверка корректности передачи: при правильно выполненной передаче в соответствующем регистре устройства управления принтера устанавливается признак завершения приема информации. наконец, принятый байт обрабатывается принтером - выполняется соответствующая команда или печатается символ.

Обязанности между драйвером и контроллером могут распределяться по-разному, но чаще всего контроллер поддерживает набор простых команд, служащих для управления периферийным устройством, а на драйвер конечно возлагают наиболее сложные функции реализации обмена. К примеру, контроллер принтера может поддерживать такие элементарные команды, как "Печать символа", "Перевод строки", "возврат каретки" и т.п.

Драйвер же принтера с помощью этих команд реализует печать строк символов, разделение документа на страницы и другие более высокоуровневые операции (например, подсчет контрольной суммы последовательности передаваемых байтов, анализ периферийного устройства, проверка правильности выполнения команды). Драйвер, задавая ту или иную последовательность команд, определяет

тем самым логику работы периферийного устройства. Для того же контроллера можно разработать различные драйверы, которые с помощью того же набора доступных команд будут реализовывать различные алгоритмы управления тем самым ПП.

Возможен распределение функций между драйвером и контроллером (ПП).

Функции, выполняемые драйвером:

- ведение очередей запросов;
- буферизация данных;
- подсчет контрольной суммы последовательности байтов;
- анализ ПП;
- загрузки очередного байта данных (или команды) в регистр контроллера;
- считывания байта данных или байта состояния ЧП из регистра контроллера.

Функции, выполняемые контроллером:

- преобразования байта из регистра (порта) в последовательность бит;
- передача каждого бита в линию связи;
- обрамление б стартовым и стоповым битами - синхронизация;
- формирования бита четности;
- установка признака завершения приема / передачи байта.

Связь двух компьютеров

А теперь предположим, что пользователь другого компьютера хотел бы распечатать текст.

Сложность заключается в том, что к его компьютеру не подключен принтер, и нужно воспользоваться тем принтером, который связан с другим компьютером.

Программа, работающая на одном компьютере, не может получить непосредственный доступ к ресурсам другого компьютера - его дискам, файлам, принтеру. Она может только «попросить» о этот другую программу, выполняемую на том компьютере, которому принадлежат эти ресурсы. Эти "просьбы" выражаются в виде сообщений, передаваемых по каналам связи между компьютерами. такая организация печати называется удаленной.

Предположим, что мы связали компьютеры по кабелю через Com-порта, которые, как известно, реализуют интерфейс RS-232C (такое соединение часто называют нуль-модемным). связь между компьютерами осуществляется аналогично связи компьютера с ЧП. Только теперь контроллеры и драйверы портов действуют с двух сторон. Вместе они обеспечивают передачу по кабелю между компьютерами одного байта информации. (В "настоящих" локальных сетях подобные функции передачи данных в линию связи выполняются сетевыми адаптерами и их драйверами.)

Итак, механизм обмена байтами между двумя компьютерами определен. теперь нужно договориться о правилах обмена сообщениями между приложениями А и В. Приложение В должно "Уметь" расшифровать получаемую от приложения А информацию. Для этого программисты, разрабатывали приложения А и В, строго оговаривают форматы сообщений, которыми будут обмениваться приложения, и их семантику. Например, они могут договориться о том, что любое исполнение удаленной операции печати начинается с передачи сообщения, запрашивающего информацию о

готовность приложения В; что в начале сообщения идет число, определяющее длину данных, предназначенных для печати; что признаком срочного завершения печати есть определенная кодовая комбинация и т.п. Тем самым, как будет показано далее, определяется протокол взаимодействия приложений.

Вернемся к последовательности действий, которые необходимо выполнить для распечатки текста на принтере "чужого" компьютера.

- Приложение А формирует очередное сообщение (содержащее, например, строка, который необходимо вывести на принтер) приложения В, помещает его в буфер оперативной памяти и обращается к ОС с запросом на передачу содержимого буфера на компьютер В.

- ОС компьютера А обращается к драйверу Com-порта, который инициирует работу контроллера.

- Действующие с обеих сторон пары драйверов и контроллеров COM- порта последовательно, байт за байтом, передают сообщение на компьютер В.

- Драйвер компьютера В периодически выполняет проверку на наличие признака завершения приема, устанавливаемого контроллером при правильно выполненной передаче данных, и только он появляется считывает принятый байт из буфера контроллера в оперативную память, тем самым делая его доступным для программ компьютера В. В некоторых случаях драйвер вызывается асинхронно, по прерываниям от контроллера. Аналогично реализуется и передача байта в другую сторону - от компьютера В к компьютеру А.

- Приложение В принимает сообщение, интерпретирует его, и в зависимости от того, что в нем содержится, формирует запрос к своей ОС на выполнение тех или иных действий с принтером. В нашем примере сообщение содержит указание на печать текста, поэтому ОС передает драйверу принтера запрос на печать строки.

- Далее выполняются все действия 1-6, описывающих выполнения запроса приложения к ЧП в соответствии с рассмотренной ранее схемы "локальная ОС - драйвер ЧП - контроллер ЧП - устройство управления ЧП" (см. Предыдущий раздел). В результате строка будет напечатана.

Мы рассмотрели последовательность работы системы при передаче только одного сообщения от приложения А к приложению В. Однако порядок взаимодействия этих двух приложений может предполагать неоднократный обмен сообщениями различного типа. Например, после успешной печати строки (в предыдущем примере) согласно правилам, приложение В должно послать подтверждение ^ - подтверждение-сообщение с подтверждением. Это соответствующее сообщение приложение В помещает в буферную область оперативной памяти, а далее с помощью драйвера COM- порта передает его по канала связи в компьютер А, где оно и попадает к приложению А.

Клиент, редиректор и сервер

Можно представить, что любая программа, которой потребуется печать на "чужом" принтере, должна включать в себя функции, подобные тем, которые выполняет приложение А. Но нагружать этими стандартными действиями каждое приложение - текстовые и графические редакторы, системы управления базами данных и другие приложения - не очень рационально (хотя существует большое количество программ, которые действительно самостоятельно решают все задачи по обмену данными между компьютерами, например Kermit - программа обмена файлами через COM- порты, реализована для различных ОС, Norton Commander 3.0

с его функцией Link). Гораздо выгоднее создать специальный программный модуль, который (вместо приложения А) будет выполнять формирование уведомлений-запросов к удаленной машины и приемки результатов для всех приложений. Такой служебный модуль называется клиентом.

На стороне же компьютера В (на месте приложения В) должна работать другая специализированная программа - сервер, постоянно ожидает прихода запросов на удаленный доступ к принтеру (или файлам, расположенным на диске) этого компьютера. Сервер, приняв запрос из сети, обращается к локальному ЧП, возможно, с участием локальной ОС.

Очень удобной и полезной функцией клиентской программы является способность отличить запрос удаленного файла от запроса к локальному файлу. Если клиентская программа умеет это делать, она сама распознает и перенаправляет (redirect) запрос к удаленной машины. Отсюда и название, часто используемое для клиентской части - редиректор. Иногда функции распознавания выделяются в особый программный модуль, в этом случае редиктором называют не всю клиентскую часть, а только этот модуль.

Программные клиент и сервер выполняют системные функции по обслуживанию запросов всех приложений компьютера А на удаленный доступ к файлам компьютера В. Чтобы приложения компьютера В могли пользоваться файлами компьютера А, описанную схему нужно симметрично дополнить клиентом для компьютера В и сервером для компьютера А. Схема взаимодействия клиента и сервера с приложениями и локальной операционной системой.

Для того, чтобы компьютер мог работать в сети, его операционная система должна быть дополнена клиентским и / или серверным модулем, а также средствами передачи данных между компьютерами. В результате такого добавления операционная система компьютера становится сетевой ОС.

Задача физической передачи данных по линиям связи

Даже при рассмотрении простейшей сети, состоит всего из двух машин, можно увидеть многие проблемы, присущие любой вычислительной сети, в том числе, связанные с **физической передачей** сигналов по линиям связи.

При соединении "точка-точка" на первый план выходит задача физической передачи данных по линиям связи. Эта задача среди прочего включает:

- **кодирования и модуляцию** данных;
- **взаимную синхронизацию** передатчика одного компьютера с приемником другого;
- подсчет **контрольной суммы** и передача ее по линиям связи после каждого байта или после

некоторого блока байтов.

В вычислительной технике для представления данных используется двоичный код. спектр данных в виде электрических или оптических сигналов называется кодированием. Существуют различные способы кодирования двоичных цифр 1 и 0, например потенциальный способ, при котором единице соответствует один уровень напряжения, а нулю - другой, или импульсный способ, когда для представления цифр используются импульсы различной или одной полярности.

Аналогичные подходы могут использоваться для кодирования данных и при их передаче между двумя компьютерами по линиям связи. Однако эти линии связи отличаются по своим электрическим

характеристиках от тех, которые существуют внутри компьютера. Главное отличие внешних линий связи от внутренних заключается в них гораздо большей длине, а также в том, что они проходят вне экранированным корпусом по просторам, чаще подданных воздействию сильных электромагнитных помех. Все это приводит к существенно большим искажениям прямоугольных импульсов (например, "Заваливание" фронтов), чем внутри компьютера. Поэтому при передаче данных внутри и вне компьютером не всегда можно использовать те же скорости и способы кодирования.

В вычислительных сетях применяют как потенциальное, так и импульсное кодирование дискретных данных, а также специфический способ представления данных, который никогда не используется внутри компьютера, - модуляцию. При модуляции дискретная информация представляется синусоидальным сигналом той частоты, которую хорошо передает имеющаяся линия связи.

Потенциальное или импульсное кодирование применяется на каналах высокого качества, а модуляция на основе синусоидальных сигналов предпочтительнее в том случае, когда канал вносит сильные искажения в передаваемые сигналы. Конечно модуляция используется в глобальных сетях при передачи данных через аналоговые телефонные линии, которые были разработаны для передачи голоса в аналоговой форме и поэтому не очень подходят для непосредственной передачи импульсов.

На способ передачи сигналов влияет и количество проводов в линиях связи между компьютерами. Чтобы снизить стоимость линий связи в сетях, разработчики пытаются сократить количество проводов и поэтому используют не параллельную передачу всех бит одного байта или даже нескольких байт, как это делается внутри компьютера, а последовательную, побитную передачу, требует всего одной пары проводов.

При передаче сигналов приходится еще решать проблему взаимной синхронизации передатчика одного компьютера с приемником другого. При организации взаимодействия модулей внутри компьютера она решается очень просто, так как в этом случае все модули синхронизируются от общего тактового генератора. Проблема синхронизации при связи компьютеров решается различными способами, как с помощью обмена специальными тактовыми синхроимпульсами по отдельной линии, так и с помощью периодической синхронизации заранее обусловленными кодами или импульсами характерной формы, отличной от формы импульсов данных.

Несмотря на принятые меры (выбор соответствующей скорости обмена данными, линий связи с определенными характеристиками, способа синхронизации приемника и передатчика), существует вероятность искривление некоторых бит передаваемых данных. Для более надежной передачи данных часто используется стандартное приема - подсчет контрольной суммы и передача ее по линиям связи после каждого байта или после некоторого блока байтов. Часто в протокол обмена данными включается как обязательный элемент сигнал-квитанция, подтверждающая правильность приема данных и посылает от получателя отправителю.

В каждый сетевой интерфейс, будь то порт маршрутизатора, концентратора или коммутатора, встроенные средства, в той или иной степени решающие задачи надежного обмена двоичными сигналами, представленными соответствующими электромагнитными сигналами. некоторые сетевые устройства, такие как модемы и сетевые адаптеры, специализирующиеся на физической передаче

данных. Модемы выполняют в глобальных сетях модуляцию и демодуляцию дискретных сигналов, синхронизируют передачу электромагнитных сигналов по линиям связи, проверяют правильность передачи по контрольной сумме и могут выполнять некоторые другие операции. сетевые адаптеры рассчитаны, как правило, на работу с определенным передаточным средой - коаксиальным кабелем, витой парой, оптоволокном и т.п. Каждый тип передающей среды имеет определенные электрические характеристики, влияющие на способ использования данной среды и определяет скорость передачи сигналов, способ их кодирования и некоторые другие параметры.