Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Амурский государственный университет»
(ФГБОУ ВПО «АмГУ»)

Кафедра ФИЗИКИ

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ

«КОМПЬЮТЕРНЫЕ СЕТИ»

Основной образовательной программы по специальности 010701.65 – «Физика»

Благовещенск 2012

УМКД разработан кан. физ.-мат. наук, доцентом кафедры ФИЗИКИ Стуковой Еленой Владимировной

Рассмотрен и рекомендован на заседании кафедры ФИЗИКИ

Протокол заседания кафедры от «___»_____2012г. №_____

И.о. зав. кафедрой

_____/_____/

(подпись) (И.О. Фамилия)

УТВЕРЖДЕН

Протокол заседания УМСС 010701.65 – «Физика»
от «___»_____2012г. №____

Председатель
УМСС_____/_____
_/

(подпись) (И.О. Фамилия)

Содержание

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Амурский государственный университет»

УТВЕРЖДАЮ
Проректор по учебной работе
_____В.В. Проказин
«_____»_____ 201__г.

РАБОЧАЯ ПРОГРАММА
Компьютерные сети
(наименование учебной дисциплины/модуля)

для специальности 010701.65 – «Физика»

по специализациям: «Информационные технологии в образовании и научной деятельности»
«Медицинская физика»

Квалификация выпускника: ФИЗИК

Курс V Семестр 9

Зачет 9
(семестр)

Лекции 28 (час.)
Самостоятельная работа 14 (час.)
Общая трудоемкость дисциплины 42 (час.)

Составитель Е.В. Стукова, доцент, канд. физ.-мат. наук.
(И.О.Ф., должность, ученое звание)

Факультет: инженерно-физический
Кафедра: теоретической и экспериментальной физики

Благовещенск 2012 г.

Рабочая программа составлена на основании Государственного образовательного стандарта высшего профессионального образования и авторских разработок по направлению подготовки 010701.65 – «Физика», квалификация: физик

Рабочая программа обсуждена и утверждена на заседании кафедры теоретической и экспериментальной физики
«___»_____ 201__г., протокол № _____
И.о. заведующего кафедрой_____ Е.А.Ванина

Рабочая программа одобрена на заседании учебно-методического совета
по направлению подготовки 010701.65 – «Физика»
«___»_____201__г., протокол №_____
Председатель УМСС_____

Рабочая программа переутверждена на заседании кафедры теоретической и экспериментальной физики от _____
протокол № _____
И.о. заведующего кафедрой_____ Е.А.Ванина

СОГЛАСОВАНО
Начальник УМУ
_____
«___»_____201__ г.

СОГЛАСОВАНО
Председатель УМС факультета
_____
«___»_____201__ г.

СОГЛАСОВАНО
Заведующий выпускающей кафедрой
_____
«___»_____201__г.

СОГЛАСОВАНО
Директор научной библиотеки
_____
«___»_____201__г.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью дисциплины является изучение принципов построения сетей, основных топологий вычислительных сетей, способов и методов передачи информации в вычислительных сетях, вопросов комплексирования сетей, ознакомление с сервисными службами локальных и глобальных сетей.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО:

Дисциплина «Компьютерные сети» является дисциплиной, входящей в блок дисциплин специализаций СД.ДС.Р.15 для специальности 010701 «Физика».

Для освоения дисциплины необходимо знать:

1) информатику;
2) архитектуру ЭВМ;
3) операционные системы.

3. ЗНАНИЯ И УМЕНИЯ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате изучения дисциплины студент должен:

знать принципы построения вычислительных сетей, основные технические средства и топологии вычислительных сетей;

уметь использовать изученные программные средства и сетевые протоколы, реализуемые ими, для решения конкретных задач;

иметь представление о перспективных направлениях развития сетевых технологий.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет _42_ часа.

№ п/п	Раздел дисциплины	Виды учебной работы		Формы текущего контроля
		Лекции (час.)	СРС (час.)	
1	ОСНОВНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ	4	2	Составление конспектов по самостоятельной работе.
2	МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ (OSI)	4	2	Составление конспектов по самостоятельной работе.
3	ФУНКЦИОНАЛЬНЫЕ ГРУППЫ УСТРОЙСТВ В СЕТИ	6	4	Составление конспектов по самостоятельной работе. Письменный опрос
4	АДРЕСАЦИЯ В СЕТЯХ. МЕЖСЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ.	4	2	Составление конспектов по самостоятельной работе.

5	КОМПЬЮТЕРНЫЕ ГЛОБАЛЬНЫЕ СЕТИ С КОММУТАЦИЕЙ ПАКЕТОВ. ИНФОРМАЦИОННЫЕ РЕСУРСЫ ИНТЕРНЕТ.	6	2	Составление конспектов по самостоятельной работе. Письменный опрос
6	БЕЗОПАСНОСТЬ И ЗАЩИТА ДАННЫХ	4	2	Составление конспектов по самостоятельной работе.
	Итого в 9-м семестре	**28**	**14**	**Зачет**

5. СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ

5.1. Лекционный курс

ТЕМА I. ОСНОВНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Эволюция вычислительных систем. Системы пакетной обработки. Многотерминальные системы - прообраз сети. Появление глобальных сетей. Первые локальные сети. Проблемы объединения нескольких компьютеров. Топология физических связей. Организация совместного использования линий связи

ТЕМА II. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ (OSI)

Понятие «открытая система» и проблемы стандартизации. Многоуровневый подход. Протокол. Интерфейс. Стек протоколов. Модель OSI. Уровни модели OSI.

ТЕМА III. ФУНКЦИОНАЛЬНЫЕ ГРУППЫ УСТРОЙСТВ В СЕТИ

Проводные и беспроводные компьютерные сети. Физическая передающая среда локальной вычислительной сети: коаксиальный кабель, витая пара, оптоволокно. Стандарты кабелей. Беспроводные каналы и их характеристики. Сетевые адаптеры. Функции и характеристики сетевых адаптеров. Классификация сетевых адаптеров. Драйверы сетевых адаптеров. Установка и конфигурирование сетевого адаптера. Коммуникационное оборудование сетей: концентраторы, мосты, коммутирующие мосты, маршрутизаторы, шлюзы, их назначение, основные функции и параметры. Модемы: назначение, виды, характеристики.

ТЕМА IV. АДРЕСАЦИЯ В СЕТЯХ. МЕЖСЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ

Адресация в IP-сетях. Форматы IP-адресов и их преобразование. Разделение сети: подсети и маски подсетей. Адресация подсетей. Реализация архитектуры подсетей. Определение маски подсети. Реализация IP-маршрутизации. Организация доменов и доменных имен. Определение имен узлов. Организация межсетевого взаимодействия. Протоколы маршрутизации. Фильтрация пакетов. Функции маршрутизатора. Сетевой шлюз. Брандмауэр.

ТЕМА V. КОМПЬЮТЕРНЫЕ ГЛОБАЛЬНЫЕ СЕТИ С КОММУТАЦИЕЙ ПАКЕТОВ. ИНФОРМАЦИОННЫЕ РЕСУРСЫ ИНТЕРНЕТ.

Организация виртуальных каналов информационного обмена. Протокол X.25. Характеристика уровней протокола. Достоинства и недостатки сетей X.25. Технология ATM (Asynchronous Transfer Mode). Протоколы уровня приложений. Различия и особенности распространенных протоколов. Согласование параметров взаимодействия. Симметрия связи «терминал-процесс». Программа-клиент Telnet. Удаленный доступ через промежуточную сеть. Электронная почта: формат, почтовые клиенты, протоколы. Протоколы распределенных файловых систем: FTP, Gopher, NNTP. Протокол пересылки гипертекста HTTP. Web-браузеры.

ТЕМА VI. БЕЗОПАСНОСТЬ И ЗАЩИТА ДАННЫХ

Обеспечение безопасности данных. Методы, средства и механизмы защиты данных. Методы, средства и механизмы безопасности данных.

6. САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ

В течение семестра студентами должны быть самостоятельно изучены следующие вопросы и подготовлен реферат по заданной теме:

№ п/п	Форма самостоятельной работы	Кол-во уч. часов
1	Составление конспекта на тему «Администрирование компьютерных сетей»	2
2	Составление конспекта на тему «Передача данных в компьютерных сетях»	2
3	Составление конспекта на тему «Физическое и логическое кодирование данных»	2
4	Составление конспекта на тему «Беспроводные компьютерные сети»	2
5	Составление конспекта на тему «Контроль передачи данных в компьютерных сетях»	2
6	Подготовка реферата на одну из заданных тем: 1. Составные компьютерные сети 2. Иерархическая организация компьютерных сетей 3. Систематизация методов доступа в среду передачи данных 4. Организация удаленного доступа в компьютерных сетях 5. Стек протоколов TCP/IP 6. Коммутация каналов и пакетов в компьютерных сетях 7. Использование иерархии цифровых выделенных линий PDH 8. Иерархия цифровых выделенных линий SONET/SDH 9. Мониторинг в компьютерных сетях 10. Управление в компьютерных сетях 11. Видеоконференции в компьютерных сетях 12. Организация Непосредственного общения через компьютерную сеть 13. Иерархия средств телекоммуникации в компьютерных сетях 14. Пакетный мобильный радиосервис в компьютерных сетях (GPRS) 15. Мультимедиа и компьютерные сети	4
	Всего	14

7. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Вид инноваций	Перечень инноваций
1. Методы, применяемые в обучении	Неимитационные методы обучения: *проблемная лекция, лекция-консультация.* Неигровые имитационные методы обучения: *контекстное обучение, метод решения творческих задач* (применяется в ходе практических занятий); *кейс-метод* (используется в ходе лабораторных занятий).
2. Технологии обучения	Компетентностно-ориентированное обучение
3. Информационные технологии	Лекции проводятся с использованием интерактивной доски и мультимедийного оборудования.
4. Информационные системы	Электронный ресурс библиотеки АмГУ: http://www.biblio@amursu.ru/**.**
5. Инновационные методы контроля	Компьютерное интернет-тестирование.

8. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

8.1. Примерные вопросы к зачету

1. Системы пакетной обработки. Многотерминальные системы.
2. Многоуровневый подход. Интерфейс. Протокол. Стек протоколов.
3. Модель взаимодействия открытых систем. Принципы передачи информации.
4. Физический и канальный уровни МВОС.
5. Сетевой, транспортный, представительный и прикладной уровни МВОС.
6. Кабели на основе витых пар.
7. Коаксиальные кабели.
8. Оптоволокно.
9. Беспроводные линии связи.
10. Код передачи данных NRZ
11. Код передачи данных RZ
12. Манчестерский код передачи данных.
13. Бифазный код передачи данных.
14. Коды передачи данных NRZI MLT-3.
15. кодирование.
16. IP-адресация.
17. Стек протоколов TCP/IP.

Текущий контроль за аудиторной и самостоятельной работой обучаемых осуществляется во время проведения аудиторных занятий посредством устного опроса, проведения контрольных работ или осуществления лекции в форме диалога.

Промежуточный контроль осуществляется один раз в семестр в виде анализа разработанных логических схем построения локальных вычислительных сетей.

Зачет – итоговый контроль осуществляется после успешного прохождения студентами текущего и промежуточного контроля в виде устного или письменного зачета при ответах на два вопроса в билете и дополнительные вопросы по желанию преподавателя.

Зачтено – изложение полученных знаний в устной, письменной или графической форме, полное, в системе, в соответствии с требованиями учебной программы; допускаются единичные несущественные ошибки, самостоятельно исправляемые студентами; допускаются отдельные несущественные ошибки, исправляемые студентами после указания преподавателя на них; допускаются отдельные существенные ошибки, исправление с помощью преподавателя.

Не зачтено – изложение учебного материала неполное, бессистемное, что препятствует усвоению последующей учебной информации; существенные ошибки, неисправляемые даже с помощью преподавателя.

9. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ «КОМПЬЮТЕРНЫЕ СЕТИ»

а) основная литература:

1. **Мельников, В. П.**Информационные технологии [Текст] : учеб. пособие : рек. УМО / В.П. Мельников. - М.: Академия, 2008. – 426с.
2. **Таненбаум, Э**. Компьютерные сети [Текст] / Э. Таненбаум. - 4-е изд. - СПб.: Питер, 2009, 2010. - 992 с.
3. **Антонова, Г. М**. Современные средства ЭВМ и телекоммуникаций [Текст] : учеб. пособие : рек. НМС / Г. М. Антонова, А. Ю. Байков. - М.: Академия, 2010. - 144 с.

б) дополнительная литература:

1. **Шапорев, С. Д**. Информатика. Теоретический курс и практические занятия [Текст]: учеб.: рек. НМС / С. Д. Шапорев. - СПб.: БХВ-Петербург, 2008. - 469 с.
2. **Информатика** [Текст] : учеб: рек. Мин. обр. РФ / под ред. Н. В. Макаровой. - 3-е изд., перераб. - М.: Финансы и статистика, 2001, 2005, 2007, 2009. - 768 с.

в) периодическая литеретура:

- Журнал «Информационные технологии и вычислительные системы»
- Журнал «Информационные системы и технологии».

г) программное обеспечение и Интернет-ресурсы:

№	Наименование ресурса	Краткая характеристика
1	www.pitbooks.ru/seti/	Сайт бесплатных электронных книг. Некоммерческий проект, создан с целью оказания помощи школьникам и студентам в изучении физики и других предметов. На этом ресурсе размещены различные материалы: учебники, задачники, лекции, другие учебные пособия. Все выложенные материалы для вас бесплатны и при скачивания не требуют каких-либо регистраций.
2	Свободная энциклопедия Википедия	Интернет-энциклопедия образовательных изданий, в которой собраны электронные учебники,

	http://ru.wikipedia.	справочники, а так же статьи различной тематики. Удобный поиск по ключевым словам, отдельным темам и отраслям знания.
3	Электронная библиотечная система **«Университетская библиотека- online»** www.biblioclub.ru	ЭБС по тематике охватывает всю область естественно-научных знаний и предназначена для использования в процессе обучения в высшей школе, как студентами так и преподавателями.

10. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Комплект ТСО

1. Интерактивная доска
2. Видеопроектор Epson
3. Мультимедийный проектор-03г
4. Ноутбук Пентиум 100-03г.

Программа составлена в соответствии с требованиями ГОС ВПО с учетом рекомендаций и ПрООП ВПО по специальности.

Тема 1. Основные принципы построения вычислительных сетей

Эволюция вычислительных систем

Концепция вычислительных сетей является логическим результатом эволюции компьютерной технологии. Первые компьютеры 50-х годов - большие, громоздкие и дорогие - предназначались для очень небольшого числа избранных пользователей. Часто эти монстры занимали целые здания. Такие компьютеры не были предназначены для интерактивной работы пользователя, а использовались в режиме пакетной обработки.

Системы пакетной обработки

Системы пакетной обработки, как правило, строились на базе мэйнфрейма - мощного и надежного компьютера универсального назначения. Пользователи подготавливали перфокарты, содержащие данные и команды программ, и передавали их в вычислительный центр. Операторы вводили эти карты в компьютер, а распечатанные результаты пользователи получали обычно только на следующий день (рис. 1.). Таким образом, одна неверно набитая карта означала как минимум суточную задержку.

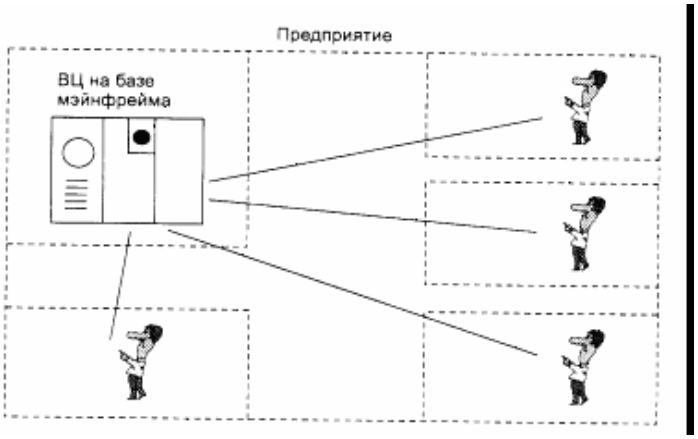


Рис. 1. Централизованная система на базе мэйнфрейма

Конечно, для пользователей интерактивный режим работы, при котором можно с терминала оперативно руководить процессом обработки своих данных, был бы гораздо удобней. Но интересами пользователей на первых этапах развития вычислительных систем в значительной степени пренебрегали, поскольку пакетный режим - это самый эффективный режим использования вычислительной мощности, так как он позволяет выполнить в

единицу времени больше пользовательских задач, чем любые другие режимы. Во главу угла ставилась эффективность работы самого дорогого устройства вычислительной машины - процессора, в ущерб эффективности работы использующих его специалистов.

Многотерминальные системы - прообраз сети

По мере удешевления процессоров в начале 60-х годов появились новые способы организации вычислительного процесса, которые позволили учесть интересы пользователей. Начали развиваться интерактивные многотерминальные системы разделения времени (рис. 2). В таких системах компьютер отдавался в распоряжение сразу нескольким пользователям. Каждый пользователь получал в свое распоряжение терминал, с помощью которого он мог вести диалог с компьютером. Причем время реакции вычислительной системы было достаточно мало для того, чтобы пользователю была не слишком заметна параллельная работа с компьютером и других пользователей. Разделяя, таким образом компьютер, пользователи получили возможность за сравнительно небольшую плату пользоваться преимуществами компьютеризации.

Терминалы, выйдя за пределы вычислительного центра, рассредоточились по всему предприятию. И хотя вычислительная мощность оставалась полностью централизованной, некоторые функции - такие как ввод и вывод данных - стали распределенными. Такие многотерминальные централизованные системы внешне уже были очень похожи на локальные вычислительные сети. Действительно, рядовой пользователь работу за терминалом мэйнфрейма воспринимал примерно так же, как сейчас он воспринимает работу за подключенным к сети персональным компьютером. Пользователь мог получить доступ к общим файлам и периферийным устройствам, при этом у него поддерживалась полная иллюзия единоличного владения компьютером, так как он мог запустить нужную ему программу в любой момент и почти сразу же получить результат. (Некоторые, далекие от

вычислительной техники пользователи даже были уверены, что все вычисления выполняются внутри их дисплея.)

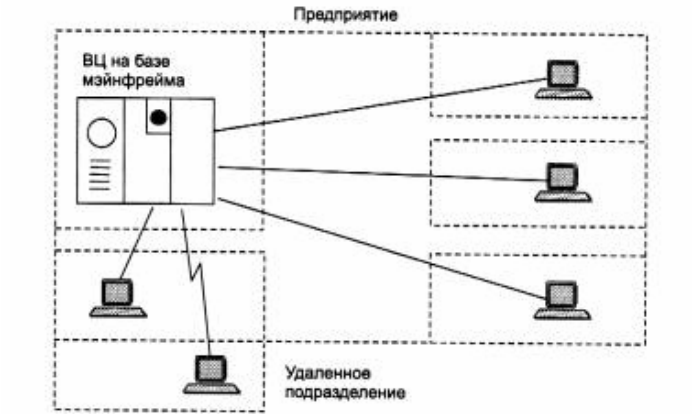


Рис. 2. Многотерминальная система - прообраз вычислительной сети

Появление глобальных сетей

Тем не менее, потребность в соединении компьютеров, находящихся на большом расстоянии друг от друга, к этому времени вполне назрела. Началось все с решения более простой задачи - доступа к компьютеру с терминалов, удаленных от него на многие сотни, а то и тысячи километров. Терминалы соединялись с компьютерами через телефонные сети с помощью модемов. Такие сети позволяли многочисленным пользователям получать удаленный доступ к разделяемым ресурсам нескольких мощных компьютеров класса суперЭВМ. Затем появились системы, в которых наряду с удаленными соединениями типа терминал-компьютер были реализованы и удаленные связи типа компьютер-компьютер. Компьютеры получили возможность обмениваться данными в автоматическом режиме, что, собственно, и является базовым механизмом любой вычислительной сети. Используя этот механизм, в первых сетях были реализованы службы обмена файлами, синхронизации баз данных, электронной почты и другие, ставшие теперь традиционными сетевые службы.

Таким образом, хронологически первыми появились глобальные вычислительные сети. Именно при построении глобальных сетей были впервые предложены и отработаны многие основные идеи и концепции современных вычислительных сетей. Такие, например, как многоуровневое

построение коммуникационных протоколов, технология коммутации пакетов, маршрутизация пакетов в составных сетях.

Первые локальные сети

В начале 70-х годов произошел технологический прорыв в области производства компьютерных компонентов - появились большие интегральные схемы. Их сравнительно невысокая стоимость и высокие функциональные возможности привели к созданию мини-компьютеров, которые стали реальными конкурентами мэйнфреймов. Закон Гроша перестал соответствовать действительности, так как десяток мини-компьютеров выполнял некоторые задачи (как правило, хорошо распараллеливаемые) быстрее одного мэйнфрейма, а стоимость такой мини-компьютерной системы была меньше.

Даже небольшие подразделения предприятий получили возможность покупать для себя компьютеры. Мини-компьютеры выполняли задачи управления технологическим оборудованием, складом и другие задачи уровня подразделения предприятия. Таким образом, появилась концепция распределения компьютерных ресурсов по всему предприятию. Однако при этом все компьютеры одной организации по-прежнему продолжали работать автономно (рис. 3).

Но шло время, потребности пользователей вычислительной техники росли, им стало недостаточно собственных компьютеров, им уже хотелось получить возможность обмена данными с другими близко расположенными компьютерами. В ответ на эту потребность предприятия и организации стали соединять свои мини-компьютеры вместе и разрабатывать программное обеспечение, необходимое для их взаимодействия. В результате появились первые локальные вычислительные сети (рис. 4). Они еще во многом отличались от современных локальных сетей, в первую очередь - своими устройствами сопряжения. На первых порах для соединения компьютеров друг с другом использовались самые разнообразные нестандартные устройства со своим способом представления данных на линиях связи,

своими типами кабелей и т. п. Эти устройства могли соединять только те типы компьютеров, для которых были разработаны, - например, мини-компьютеры PDP-11 с мэйнфреймом IBM 360 или компьютеры «Наири» с компьютерами «Днепр». Такая ситуация создала большой простор для творчества студентов - названия многих курсовых и дипломных проектов начинались тогда со слов «Устройство сопряжения...».

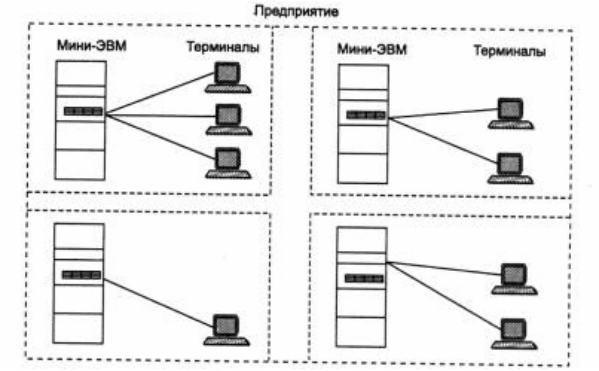


Рис. 3. Автономное использование нескольких мини-компьютеров на одном предприятии

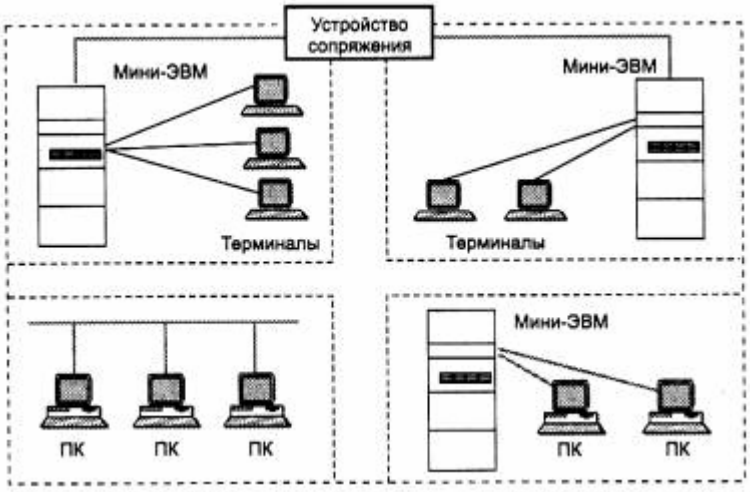
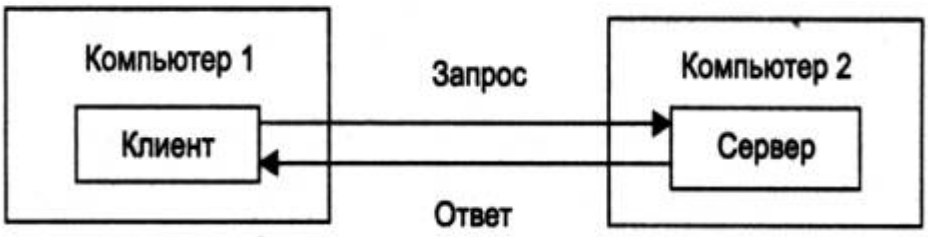


Рис. 4. Различные типы связей в первых локальных сетях.

Сетевые службы всегда представляют собой распределенные программы. *Распределенная программа* - это программа, которая состоит из нескольких взаимодействующих частей (в приведенном на рис. 5 примере - из двух), причем каждая часть, как правило, выполняется на отдельном компьютере сети.

Рис. 

5.

Взаимодействие частей распределенного приложения

До сих пор речь шла о системных распределенных программах. Однако в сети могут выполняться и распределенные пользовательские программы - приложения. Распределенное приложение также состоит из нескольких частей, каждая и которых выполняет какую-то определенную законченную работу по решению прикладной задачи. Например, одна часть приложения, выполняющаяся на компьютере пользователя, может поддерживать специализированный графический интерфейс вторая - работать на мощном выделенном компьютере и заниматься статистической обработкой введенных пользователем данных, а третья - заносить полученные результаты в базу данных на компьютере с установленной стандартной СУБД. Распределенные приложения в полной мере используют потенциальные возможности распределенной обработки, предоставляемые вычислительной сетью, и поэтому часто называются *сетевыми приложениями*.

Следует подчеркнуть, что не всякое приложение, выполняемое в сети, является сетевым. Существует большое количество популярных приложений, которые не являются распределенными и целиком выполняются на одном компьютере сети. Тем не менее и такие приложения могут использовать преимущества сети за счет встроенных в операционную систему сетевых служб. Значительная часть истории локальных сетей связана как раз с использованием таких нераспределенных приложений. Рассмотрим, например, как происходила работа пользователя с известной в свое время СУБД dBase. Обычно файлы базы данных, с которыми работали все пользователи сети, располагались на файловом сервере. Сама же СУБД

хранилась на каждом клиентском компьютере в виде единого программного модуля.

Программа dBase была рассчитана на обработку только локальных данных, то есть данных, расположенных на том же компьютере, что и сама программа. Пользователь запускал dBase на своем компьютере, и она искала данные на локальном диске, совершенно не принимая во внимание существование сети. Чтобы обрабатывать с помощью dBase данные на удаленном компьютере, пользователь обращался к услугам файловой службы, которая доставляла данные с сервера на клиентский компьютер и создавала для СУБД эффект их локального хранения.

Большинство приложений, используемых в локальных сетях в середине 80-х годов, являлись обычными, нераспределенными приложениями. И это понятно - они были написаны для автономных компьютеров, а потом просто были перенесены в сетевую среду. Создание же распределенных приложений, хотя и сулило много преимуществ (уменьшение сетевого трафика, специализация компьютеров), оказалось делом совсем не простым. Нужно было решать множество дополнительных проблем - на сколько частей разбить приложение, какие функции возложить на каждую часть, как организовать взаимодействие этих частей, чтобы в случае сбоев и отказов оставшиеся части корректно завершали работу, и т. д., и т. п. Поэтому до сих пор только небольшая часть приложений является распределенными, хотя очевидно, что именно за этим классом приложений будущее, так как они в полной мере могут использовать потенциальные возможности сетей по распараллеливанию вычислений.

Проблемы объединения нескольких компьютеров

При объединении в сеть большего числа компьютеров возникает целый комплекс новых проблем.

Топология физических связей

В первую очередь необходимо выбрать способ организации физических связей, то есть *топологию*. Под топологией вычислительной сети

понимается конфигурация графа, вершинам которого соответствуют компьютеры сети (иногда и другое оборудование, например концентраторы), а ребрам - физические связи между ними. Компьютеры, подключенные к сети, часто называют *станциями* или *узлами* сети.

Заметим, что конфигурация *физических связей* определяется электрическими соединениями компьютеров между собой и может отличаться от конфигурации *логических связей* между узлами сети. Логические связи представляют собой маршруты передачи данных между узлами сети и образуются путем соответствующей настройки коммуникационного оборудования.

Выбор топологии электрических связей существенно влияет на многие характеристики сети. Например, наличие резервных связей повышает надежность сети и делает возможным балансирование загрузки отдельных каналов. Простота присоединения новых узлов, свойственная некоторым топологиям, делает сеть легко расширяемой. Экономические соображения часто приводят к выбору топологий, для которых характерна минимальная суммарная длина линий связи. Рассмотрим некоторые, наиболее часто встречающиеся топологии.

Полносвязная топология (рис. 6, а) соответствует сети, в которой каждый компьютер сети связан со всеми остальными. Несмотря на логическую простоту, этот вариант оказывается громоздким и неэффективным. Действительно, каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров сети. Для каждой пары компьютеров должна быть выделена отдельная электрическая линия связи. Полносвязные топологии применяются редко, так как не удовлетворяют ни одному из приведенных выше требований. Чаще этот вид топологии используется в многомашинных комплексах или глобальных сетях при небольшом количестве компьютеров.

Все другие варианты основаны на неполносвязных топологиях, когда для обмена данными между двумя компьютерами может потребоваться промежуточная передача данных через другие узлы сети.

Ячеистая топология (*mesh*) получается из полносвязной путем удаления некоторых возможных связей (рис. 6, б). В сети с ячеистой топологией непосредственно связываются только те компьютеры, между которыми происходит интенсивный обмен данными, а для обмена данными между компьютерами, не соединенными прямыми связями, используются транзитные передачи через промежуточные узлы. Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для глобальных сетей.

Общая шина (рис. 6, в) является очень распространенной (а до недавнего времени самой распространенной) топологией для локальных сетей. В этом случае компьютеры подключаются к одному коаксиальному кабелю по схеме «монтажного ИЛИ». Передаваемая информация может распространяться в обе стороны. Применение общей шины снижает стоимость проводки, унифицирует подключение различных модулей, обеспечивает возможность почти мгновенного широковещательного обращения ко всем станциям сети. Таким образом, основными преимуществами такой схемы являются дешевизна и простота разводки кабеля по помещениям. Самый серьезный недостаток общей шины заключается в ее низкой надежности: любой дефект кабеля или какого-нибудь из многочисленных разъемов полностью парализует всю сеть. К сожалению, дефект коаксиального разъема редкостью не является. Другим недостатком общей шины является ее невысокая производительность, так как при таком способе подключения в каждый момент времени только один компьютер может передавать данные в сеть. Поэтому пропускная способность канала связи всегда делится здесь между всеми узлами сети.

Топология *звезда* (рис. 6, г). В этом случае каждый компьютер подключается отдельным кабелем к общему устройству, называемому

концентратором, который находится в центре сети. В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. Главное преимущество этой топологии перед общей шиной - существенно большая надежность. Любые неприятности с кабелем касаются лишь того компьютера, к которому этот кабель присоединен, и только неисправность концентратора может вывести из строя всю сеть. Кроме того, концентратор может играть роль интеллектуального фильтра информации, поступающей от узлов в сеть, и при необходимости блокировать запрещенные администратором передачи.

К недостаткам топологии типа звезда относится более высокая стоимость сетевого оборудования из-за необходимости приобретения концентратора. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора. Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой связями типа звезда (рис. 6, д). В настоящее время иерархическая звезда является самым распространенным типом топологии связей как в локальных, так и глобальных сетях.

В сетях с *кольцевой* конфигурацией (рис. 6, е) данные передаются по кольцу от одного компьютера к другому, как правило, в одном направлении. Если компьютер распознает данные как «свои», то он копирует их себе во внутренний буфер. В сети с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какой-либо станции не прервался канал связи между остальными станциями. Кольцо представляет собой очень удобную конфигурацию для организации обратной связи - данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому этот узел может контролировать процесс доставки данных адресату. Часто это свойство кольца используется для тестирования связности сети и поиска узла, работающего некорректно. Для этого в сеть посылаются специальные тестовые сообщения.

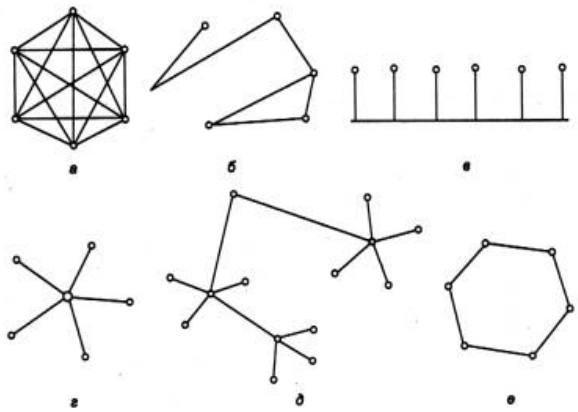


Рис. 6. Типовые топологии сетей

В то время как небольшие сети, как правило, имеют типовую топологию - звезда, кольцо или общая шина, для крупных сетей характерно наличие произвольных связей между компьютерами. В таких сетях можно выделить отдельные произвольно связанные фрагменты (подсети), имеющие типовую топологию, поэтому их называют сетями со *смешанной топологией* (рис. 7).

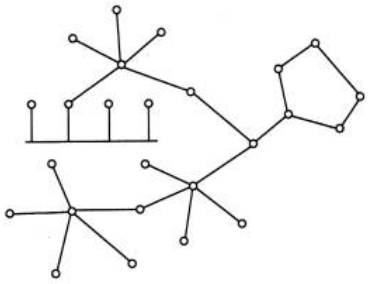


Рис. 7. Смешанная топология

Организация совместного использования линий связи

Только в сети с полносвязной топологией для соединения каждой пары компьютеров имеется отдельная линия связи. Во всех остальных случаях неизбежно возникает вопрос о том, как организовать совместное использование линий связи несколькими компьютерами сети. Как и всегда при разделении ресурсов, главной целью здесь является удешевление сети.

В вычислительных сетях используют как индивидуальные линии связи между компьютерами, так и *разделяемые (shared)*, когда одна линия связи попеременно используется несколькими компьютерами. В случае применения разделяемых линий связи (часто используется также термин

разделяемая среда передачи данных - shared media) возникает комплекс проблем, связанных с их совместным использованием, который включает как чисто электрические проблемы обеспечения нужного качества сигналов при подключении к одному и тому же проводу нескольких приемников и передатчиков, так и логические проблемы разделения во времени доступа к этим линиям.

Классическим примером сети с разделяемыми линиями связи являются сети с топологией «общая шина», в которых один кабель совместно используется всеми компьютерами сети. Ни один из компьютеров сети в принципе не может индивидуально, независимо от всех других компьютеров сети, использовать кабель, так как при одновременной передаче данных сразу несколькими узлами сигналы смешиваются и искажаются. В топологиях «кольцо» или «звезда» индивидуальное использование линий связи, соединяющих компьютеры, принципиально возможно, но эти кабели часто также рассматривают как разделяемые для всех компьютеров сети, так что, например, только один компьютер кольца имеет право в данный момент времени отправлять по кольцу пакеты другим компьютерам.

Существуют различные способы решения задачи организации совместного доступа к разделяемым линиям связи. Внутри компьютера проблемы разделения линий связи между различными модулями также существуют - примером является доступ к системной шине, которым управляет либо процессор, либо специальный арбитр шины. В сетях организация совместного доступа к линиям связи имеет свою специфику из-за существенно большего времени распространения сигналов по длинным проводам, к тому же это время для различных пар компьютеров может быть различным. Из-за этого процедуры согласования доступа к линии связи могут занимать слишком большой промежуток времени и приводить к значительным потерям производительности сети.

Несмотря на все эти сложности, в локальных сетях разделяемые линии связи используются очень часто. Этот подход, в частности, реализован в

широко распространенных классических технологиях Ethernet и Token Ring. Однако в последние годы наметилась тенденция отказа от разделяемых сред передачи данных и в локальных сетях. Это связано с тем, что за достигаемое таким образом удешевление сети приходится расплачиваться производительностью.

Сеть с разделяемой средой при большом количестве узлов будет работать всегда медленнее, чем аналогичная сеть с индивидуальными линиями связи, так как пропускная способность индивидуальной линии связи достается одному компьютеру, а при ее совместном использовании - делится на все компьютеры сети.

Часто с такой потерей производительности мирятся ради увеличения экономической эффективности сети. Не только в классических, но и в совсем новых технологиях, разработанных для локальных сетей, сохраняется режим разделяемых линий связи. Например, разработчики технологии Gigabit Ethernet, принятой в 1998 году в качестве нового стандарта, включили режим разделения передающей среды в свои спецификации наряду с режимом работы по индивидуальным линиям связи.

При использовании индивидуальных линий связи в полносвязных топологиях конечные узлы должны иметь по одному порту на каждую линию связи. В звездообразных топологиях конечные узлы могут подключаться индивидуальными линиями связи к специальному устройству - коммутатору. В глобальных сетях коммутаторы использовались уже на начальном этапе, а в локальных сетях - с начала 90-х годов. Коммутаторы приводят к существенному удорожанию локальной сети, поэтому пока их применение ограничено, но по мере снижения стоимости коммутации этот подход, возможно, вытеснит применение разделяемых линий связи. Необходимо подчеркнуть, что индивидуальными в таких сетях являются только линии связи между конечными узлами и коммутаторами сети, а связи между коммутаторами остаются разделяемыми, так как по ним передаются сообщения разных конечных узлов (рис. 8).

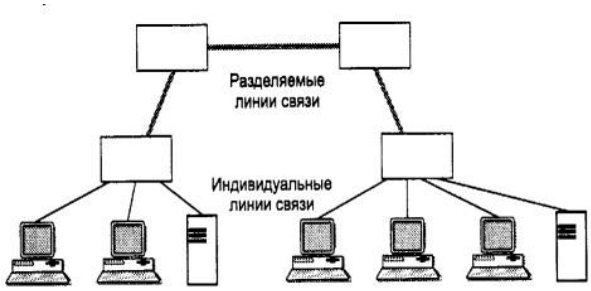


Рис. 8. Индивидуальные и разделяемые линии связи в сетях на основе коммутаторов

В глобальных сетях отказ от разделяемых линий связи объясняется техническими причинами. Здесь большие временные задержки распространения сигналов принципиально ограничивают применимость техники разделения линии связи. Компьютеры могут затратить больше времени на переговоры о том, кому сейчас можно использовать линию связи, чем непосредственно на передачу данных по этой линии связи. Однако это не относится к линиям связи типа «коммутатор - коммутатор». В этом случае только два коммутатора борются за доступ к линии связи, и это существенно упрощает задачу организации совместного использования линии.

Тема 2. Модель взаимодействия открытых сетей (OSI)

Понятие «открытая система» и проблемы стандартизации

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.

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). 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.

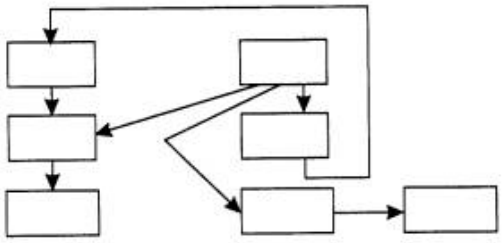


Рис. 1. Пример декомпозиции задачи

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. 2). 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.

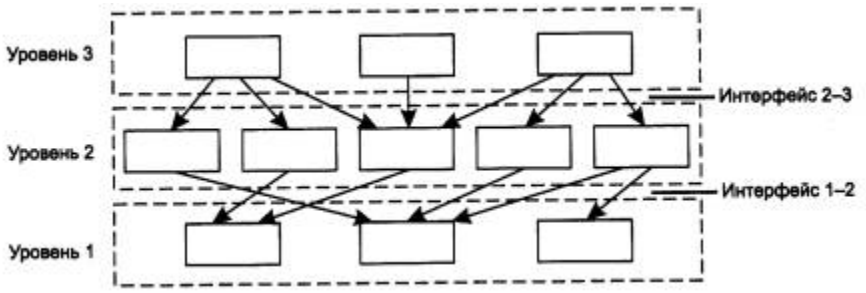


Рис. 2. Многоуровневый подход - создание иерархии задач

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. 3). Запрос вначале анализируется верхним уровнем, на котором осуществляется последовательный разбор составного символьного имени файла и определение уникального идентификатора файла. Следующий уровень находит по уникальному имени все основные характеристики файла: адрес, атрибуты доступа и т. п. 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.

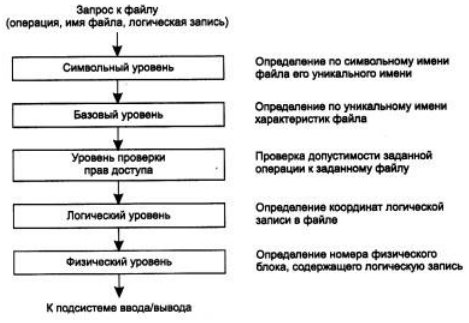


Рис. 3. Многоуровневая модель файловой системы

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. 4 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*.

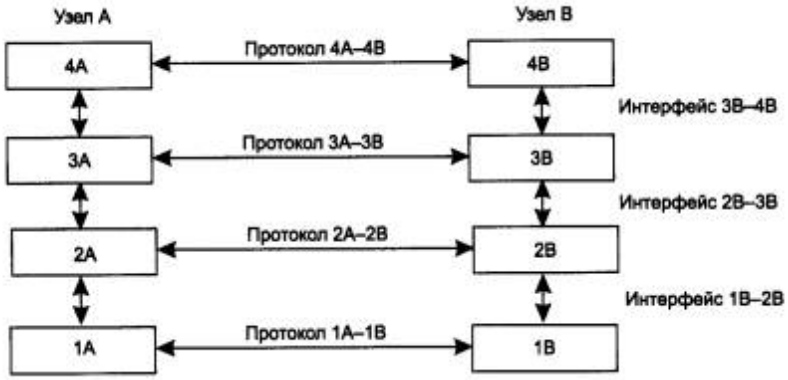


Рис. 4. Взаимодействие двух узлов

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. 2.5 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.

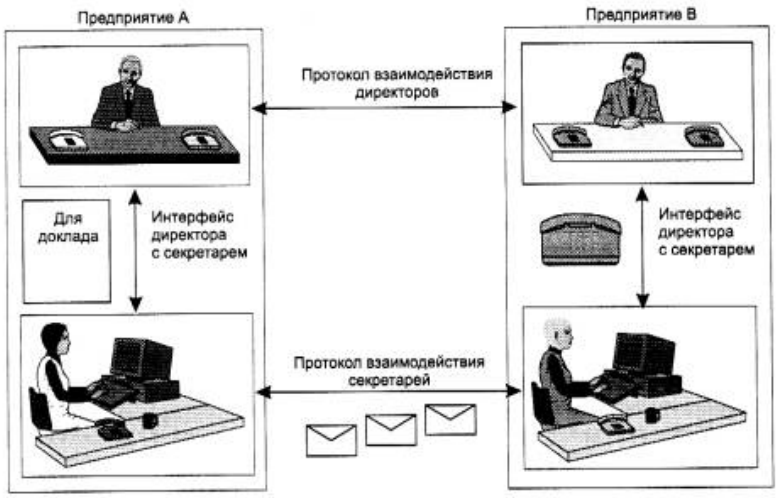


Рис. 5. Пример многоуровневого взаимодействия предприятий

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.6) 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.

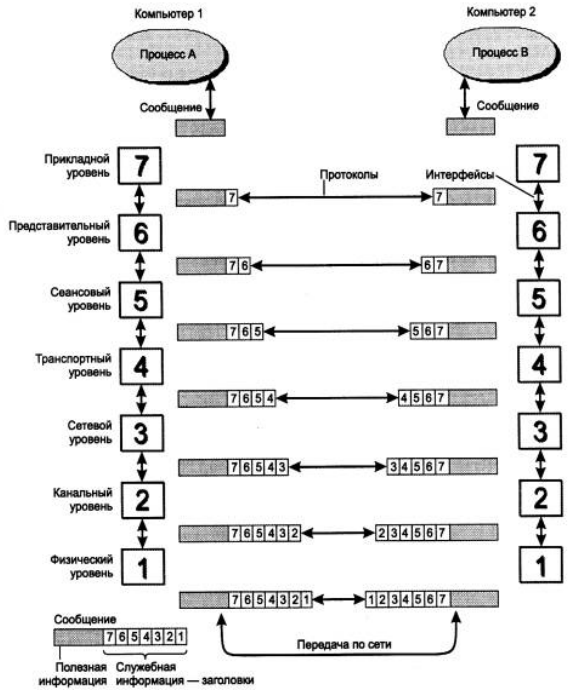


Рис. 6. Модель взаимодействия открытых систем ISO/OSI

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. 7).

Рис. 7. 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.

Уровни модели OSI

Физический уровень

Физический уровень (Physical layer) имеет дело с передачей битов по физическим каналам связи, таким, например, как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. К этому уровню имеют отношение характеристики физических сред передачи данных, такие как полоса пропускания, помехозащищенность, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, передающих дискретную информацию, например, крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизуются типы разъемов и назначение каждого контакта.

Функции физического уровня реализуются во всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Примером протокола физического уровня может служить спецификация l0-Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического

сегмента 100 метров, манчестерский код для представления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов.

Канальный уровень

На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня (Data Link layer) является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые *кадрами (frames)*. Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит в начало и конец каждого кадра, для его выделения, а также вычисляет контрольную сумму, обрабатывая все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит по сети, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка. Канальный уровень может не только обнаруживать ошибки, но и исправлять их за счет повторной передачи поврежденных кадров. Необходимо отметить, что функция исправления ошибок не является обязательной для канального уровня, поэтому в некоторых протоколах этого уровня она отсутствует, например, в Ethernet и frame relay.

В протоколах канального уровня, используемых в локальных сетях, заложена определенная структура связей между компьютерами и способы их адресации. Хотя канальный уровень и обеспечивает доставку кадра между любыми двумя узлами локальной сети, он это делает только в сети с совершенно определенной топологией связей, именно той топологией, для которой он был разработан. К таким типовым топологиям, поддерживаемым

протоколами канального уровня локальных сетей, относятся общая шина, кольцо и звезда, а также структуры, полученные из них с помощью мостов и коммутаторов. Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI, l00VG-AnyLAN.

В локальных сетях протоколы канального уровня используются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

В глобальных сетях, которые редко обладают регулярной топологией, канальный уровень часто обеспечивает обмен сообщениями только между двумя соседними компьютерами, соединенными индивидуальной линией связи. Примерами протоколов «точка-точка» (как часто называют такие протоколы) могут служить широко распространенные протоколы PPP и LAP-B. В таких случаях для доставки сообщений между конечными узлами через всю сеть используются средства сетевого уровня. Именно так организованы сети X.25. Иногда в глобальных сетях функции канального уровня в чистом виде выделить трудно, так как в одном и том же протоколе они объединяются с функциями сетевого уровня. Примерами такого подхода могут служить протоколы технологий ATM и frame relay.

В целом канальный уровень представляет собой весьма мощный и законченный набор функций по пересылке сообщений между узлами сети. В некоторых случаях протоколы канального уровня оказываются самодостаточными транспортными средствами и могут допускать работу поверх них непосредственно протоколов прикладного уровня или приложений, без привлечения средств сетевого и транспортного уровней. Например, существует реализация протокола управления сетью SNMP непосредственно поверх Ethernet, хотя стандартно этот протокол работает поверх сетевого протокола IP и транспортного протокола UDP. Естественно, что применение такой реализации будет ограниченным - она не подходит для составных сетей разных технологий, например Ethernet и X.25, и даже для

такой сети, в которой во всех сегментах применяется Ethernet, но между сегментами существуют петлевид-ные связи. А вот в двухсегментной сети Ethernet, объединенной мостом, реализация SNMP над канальным уровнем будет вполне работоспособна.

Тем не менее, для обеспечения качественной транспортировки сообщений в сетях любых топологий и технологий функций канального уровня оказывается недостаточно, поэтому в модели OSI решение этой задачи возлагается на два следующих уровня - сетевой и транспортный.

Сетевой уровень

Сетевой уровень (Network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать совершенно различные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей. Функции сетевого уровня достаточно разнообразны. Начнем их рассмотрение на примере объединения локальных сетей.

Протоколы канального уровня локальных сетей обеспечивают доставку данных между любыми узлами только в сети с соответствующей типовой топологией, например топологией иерархической звезды. Это очень жесткое ограничение, которое не позволяет строить сети с развитой структурой, например, сети, объединяющие несколько сетей предприятия в единую сеть, или высоконадежные сети, в которых существуют избыточные связи между узлами. Можно было бы усложнять протоколы канального уровня для поддержания петлевидных избыточных связей, но принцип разделения обязанностей между уровнями приводит к другому решению. Чтобы с одной стороны сохранить простоту процедур передачи данных для типовых топологий, а с другой допустить использование произвольных топологий, вводится дополнительный сетевой уровень.

На сетевом уровне сам термин *сеть* наделяют специфическим значением. В данном случае под сетью понимается совокупность компьютеров, соединенных между собой в соответствии с одной из

стандартных типовых топологий и использующих для передачи данных один из протоколов канального уровня, определенный для этой топологии.

На сетевом уровне определяются два вида протоколов. Первый вид - *сетевые протоколы (routed protocols)* - реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых протоколами обмена маршрутной информацией или просто *протоколами маршрутизации (routing protocols)*. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений. Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов.

На сетевом уровне работают протоколы еще одного типа, которые отвечают за отображение адреса узла, используемого на сетевом уровне, в локальный адрес сети. Такие протоколы часто называют *протоколами разрешения адресов - Address Resolution Protocol, ARP*. Иногда их относят не к сетевому уровню, а к канальному, хотя тонкости классификации не изменяют их сути. Примерами протоколов сетевого уровня являются протокол межсетевого взаимодействия IP стека TCP/IP и протокол межсетевого обмена пакетами IPX стека Novell.

Транспортный уровень

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Транспортный уровень (Transport layer) обеспечивает приложениям или верхним уровням стека - прикладному и сеансовому - передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления

прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное - способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

Выбор класса сервиса транспортного уровня определяется, с одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких, чем транспортный, уровней, а с другой стороны, этот выбор зависит от того, насколько надежной является система транспортировки данных в сети, обеспечиваемая уровнями, расположенными ниже транспортного - сетевым, канальным и физическим. Так, например, если качество каналов передачи связи очень высокое и вероятность возникновения ошибок, не обнаруженных протоколами более низких уровней, невелика, то разумно воспользоваться одним из облегченных сервисов транспортного уровня, не обремененных многочисленными проверками, квитированием и другими приемами повышения надежности. Если же транспортные средства нижних уровней изначально очень ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок, - с помощью предварительного установления логического соединения, контроля доставки сообщений по контрольным суммам и циклической нумерации пакетов, установления тайм-аутов доставки и т. п.

Как правило, все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети - компонентами их сетевых операционных систем. В качестве примера транспортных протоколов можно привести протоколы TCP и UDP стека TCP/IP и протокол SPX стека Novell. Протоколы нижних четырех уровней обобщенно называют сетевым транспортом или транспортной подсистемой, так как они полностью решают задачу транспортировки сообщений с

заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями. Остальные три верхних уровня решают задачи предоставления прикладных сервисов на основании имеющейся транспортной подсистемы.

Сеансовый уровень

Сеансовый уровень (Session layer) обеспечивает управление диалогом: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов, хотя функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Представительный уровень

Представительный уровень (Presentation layer) имеет дело с формой представления передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC. На этом уровне может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол Secure Socket Layer (SSL), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

Прикладной уровень

Прикладной уровень (Application layer) - это в действительности просто 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. Единица данных, которой оперирует прикладной уровень, обычно называется *сообщением (message)*.

Существует очень большое разнообразие служб прикладного уровня. Приведем в качестве примера хотя бы несколько наиболее распространенных реализации файловых служб: NCP в операционной системе Novell NetWare, SMB в Microsoft Windows NT, NFS, FTP и TFTP, входящие в стек TCP/IP.

Тема 3. Функциональные группы устройств в сети

Средой передачи информации называются те линии связи (или каналы связи), по которым производится обмен информацией между компьютерами. В подавляющем большинстве компьютерных сетей (особенно локальных) используются проводные или кабельные каналы связи, хотя существуют и беспроводные сети, которые сейчас находят все более широкое применение, особенно в портативных компьютерах.

Информация в локальных сетях чаще всего передается в последовательном коде, то есть бит за битом. Такая передача медленнее и сложнее, чем при использовании параллельного кода. Однако надо учитывать то, что при более быстрой параллельной передаче (по нескольким кабелям одновременно) увеличивается количество соединительных кабелей в число раз, равное количеству разрядов параллельного кода (например, в 8 раз при 8-разрядном коде). Это совсем не мелочь, как может показаться на первый взгляд. При значительных расстояниях между абонентами сети стоимость кабеля вполне сравнима со стоимостью компьютеров и даже может превосходить ее. К тому же проложить один кабель (реже два разнонаправленных) гораздо проще, чем 8, 16 или 32. Значительно дешевле обойдется также поиск повреждений и ремонт кабеля.

Но это еще не все. Передача на большие расстояния при любом типе кабеля требует сложной передающей и приемной аппаратуры, так как при этом необходимо формировать мощный сигнал на передающем конце и детектировать слабый сигнал на приемном конце. При последовательной передаче для этого требуется всего один передатчик и один приемник. При параллельной же количество требуемых передатчиков и приемников возрастает пропорционально разрядности используемого параллельного кода. В связи с этим, даже если разрабатывается сеть незначительной длины (порядка десятка метров) чаще всего выбирают последовательную передачу.

К тому же при параллельной передаче чрезвычайно важно, чтобы длины отдельных кабелей были точно равны друг другу. Иначе в результате прохождения по кабелям разной длины между сигналами на приемном конце образуется временной сдвиг, который может привести к сбоям в работе или даже к полной неработоспособности сети. Например, при скорости передачи 100 Мбит/с и длительности бита 10 нс этот временной сдвиг не должен превышать 5—10 нс. Такую величину сдвига дает разница в длинах кабелей в 1—2 метра. При длине кабеля 1000 метров это составляет 0,1-0,2%.

Надо отметить, что в некоторых высокоскоростных локальных сетях все-таки используют параллельную передачу по 2-4 кабелям, что позволяет при заданной скорости передачи применять более дешевые кабели с меньшей полосой пропускания. Но допустимая длина кабелей при этом не превышает сотни метров. Примером может служить сегмент 100BASE-T4 сети Fast Ethernet.

Промышленностью выпускается огромное количество типов кабелей, например, только одна крупнейшая кабельная компания Belden предлагает более 2000 их наименований. Но все кабели можно разделить на три большие группы:

- электрические (медные) кабели на основе витых пар проводов (twisted pair), которые делятся на экранированные (shielded twisted pair, STP) и неэкранированные (unshielded twisted pair, UTP);

- электрические (медные) коаксиальные кабели (coaxial cable);
- оптоволоконные кабели (fibre optic).

Каждый тип кабеля имеет свои преимущества и недостатки, так что при выборе надо учитывать как особенности решаемой задачи, так и особенности конкретной сети, в том числе и используемую топологию.

Можно выделить следующие основные параметры кабелей, принципиально важные для использования в локальных сетях:

- Полоса пропускания кабеля (частотный диапазон сигналов, пропускаемых кабелем) и затухание сигнала в кабеле. Два этих параметра тесно связаны между собой, так как с ростом частоты сигнала растет затухание сигнала. Надо выбирать кабель, который на заданной частоте сигнала имеет приемлемое затухание. Или же надо выбирать частоту сигнала, на которой затухание еще приемлемо. Затухание измеряется в децибелах и пропорционально длине кабеля.
- Помехозащищенность кабеля и обеспечиваемая им секретность передачи информации. Эти два взаимосвязанных параметра показывают, как кабель взаимодействует с окружающей средой, то есть, как он реагирует на внешние помехи, и насколько просто прослушать информацию, передаваемую по кабелю.
- Скорость распространения сигнала по кабелю или, обратный параметр – задержка сигнала на метр длины кабеля. Этот параметр имеет принципиальное значение при выборе длины сети. Типичные величины скорости распространения сигнала – от 0,6 до 0,8 от скорости распространения света в вакууме. Соответственно типичные величины задержек – от 4 до 5 нс/м.
- Для электрических кабелей очень важна величина волнового сопротивления кабеля. Волновое сопротивление важно учитывать при согласовании кабеля для предотвращения отражения сигнала от концов кабеля. Волновое сопротивление зависит от формы и взаиморасположения проводников, от технологии изготовления и

материала диэлектрика кабеля. Типичные значения волнового сопротивления – от 50 до 150 Ом. В настоящее время действуют следующие стандарты на кабели:

- EIA/TIA 568 (Commercial Building Telecommunications Cabling Standard) – американский;
- ISO/IEC IS 11801 (Generic cabling for customer premises) – международный;
- CENELEC EN 50173 (Generic cabling systems) – европейский.

Эти стандарты описывают практически одинаковые кабельные системы, но отличаются терминологией и нормами на параметры. В данном курсе предлагается придерживаться терминологии стандарта EIA/TIA 568.

Кабели на основе витых пар

Витые пары проводов используются в дешевых и сегодня, пожалуй, самых популярных кабелях. Кабель на основе витых пар представляет собой несколько пар скрученных попарно изолированных медных проводов в единой диэлектрической (пластиковой) оболочке. Он довольно гибкий и удобный для прокладки. Скручивание проводов позволяет свести к минимуму индуктивные наводки кабелей друг на друга и снизить влияние переходных процессов.

Обычно в кабель входит две (рис. 1) или четыре витые пары.

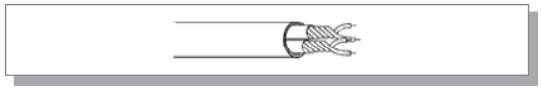


Рис. 1. Кабель с витыми парами

Неэкранированные витые пары характеризуются слабой защищенностью от внешних электромагнитных помех, а также от подслушивания, которое может осуществляться с целью, например, промышленного шпионажа. Причем перехват передаваемой по сети информации возможен как с помощью контактного метода (например, посредством двух иголок, воткнутых в кабель), так и с помощью бесконтактного метода, сводящегося к радиоперехвату излучаемых кабелем

электромагнитных полей. Причем действие помех и величина излучения во вне увеличивается с ростом длины кабеля. Для устранения этих недостатков применяется экранирование кабелей.

В случае экранированной витой пары STP каждая из витых пар помещается в металлическую оплетку-экран для уменьшения излучений кабеля, защиты от внешних электромагнитных помех и снижения взаимного влияния пар проводов друг на друга (crosstalk – перекрестные наводки). Для того чтобы экран защищал от помех, он должен быть обязательно заземлен. Естественно, экранированная витая пара заметно дороже, чем неэкранированная. Ее использование требует специальных экранированных разъемов. Поэтому встречается она значительно реже, чем неэкранированная витая пара.

Основные достоинства неэкранированных витых пар – простота монтажа разъемов на концах кабеля, а также ремонта любых повреждений по сравнению с другими типами кабеля. Все остальные характеристики у них хуже, чем у других кабелей. Например, при заданной скорости передачи затухание сигнала (уменьшение его уровня по мере прохождения по кабелю) у них больше, чем у коаксиальных кабелей. Если учесть еще низкую помехозащищенность, то понятно, почему линии связи на основе витых пар, как правило, довольно короткие (обычно в пределах 100 метров). В настоящее время витая пара используется для передачи информации на скоростях до 1000 Мбит/с, хотя технические проблемы, возникающие при таких скоростях, крайне сложны.

Согласно стандарту EIA/TIA 568, существуют пять основных и две дополнительные категории кабелей на основе неэкранированной витой пары (UTP):

- Кабель категории 1 – это обычный телефонный кабель (пары проводов не витые), по которому можно передавать только речь. Этот тип кабеля имеет большой разброс параметров (волнового сопротивления, полосы пропускания, перекрестных наводок).

- Кабель категории 2 – это кабель из витых пар для передачи данных в полосе частот до 1 МГц. Кабель не тестируется на уровень перекрестных наводок. В настоящее время он используется очень редко. Стандарт EIA/TIA 568 не различает кабели категорий 1 и 2.
- Кабель категории 3 – это кабель для передачи данных в полосе частот до 16 МГц, состоящий из витых пар с девятью витками проводов на метр длины. Кабель тестируется на все параметры и имеет волновое сопротивление 100 Ом. Это самый простой тип кабелей, рекомендованный стандартом для локальных сетей. Еще недавно он был самым распространенным, но сейчас повсеместно вытесняется кабелем категории 5.
- Кабель категории 4 – это кабель, передающий данные в полосе частот до 20 МГц. Используется редко, так как не слишком заметно отличается от категории 3. Стандартом рекомендуется вместо кабеля категории 3 переходить сразу на кабель категории 5. Кабель категории 4 тестируется на все параметры и имеет волновое сопротивление 100 Ом. Кабель был создан для работы в сетях по стандарту IEEE 802.5.
- Кабель категории 5 – в настоящее время самый совершенный кабель, рассчитанный на передачу данных в полосе частот до 100 МГц. Состоит из витых пар, имеющих не менее 27 витков на метр длины (8 витков на фут). Кабель тестируется на все параметры и имеет волновое сопротивление 100 Ом. Рекомендуется применять его в современных высокоскоростных сетях типа Fast Ethernet и TPFDDI. Кабель категории 5 примерно на 30—50% дороже, чем кабель категории 3.
- Кабель категории 6 – перспективный тип кабеля для передачи данных в полосе частот до 200 (или 250) МГц.
- Кабель категории 7 – перспективный тип кабеля для передачи данных в полосе частот до 600 МГц.

Согласно стандарту EIA/TIA 568, полное волновое сопротивление наиболее совершенных кабелей категорий 3, 4 и 5 должно составлять 100 Ом

±15% в частотном диапазоне от 1 МГц до максимальной частоты кабеля. Требования не очень жесткие: величина волнового сопротивления может находиться в диапазоне от 85 до 115 Ом. Здесь же следует отметить, что волновое сопротивление экранированной витой пары STP по стандарту должно быть равным 150 Ом ±15%. Для согласования сопротивлений кабеля и оборудования в случае их несовпадения применяют согласующие трансформаторы (Balun). Существует также экранированная витая пара с волновым сопротивлением 100 Ом, но используется она довольно редко.

Второй важнейший параметр, задаваемый стандартом, – это максимальное затухание сигнала, передаваемого по кабелю, на разных частотах. Из таблицы видно, что величины затухания на частотах, близких к предельным, для всех кабелей очень значительны. Даже на небольших расстояниях сигнал ослабляется в десятки и сотни раз, что предъявляет высокие требования к приемникам сигнала

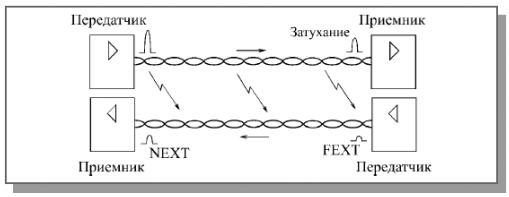


Рис. 2. Перекрестные помехи в кабелях на витых парах

Стандарт определяет также максимально допустимую величину рабочей емкости каждой из витых пар кабелей категории 4 и 5. Она должна составлять не более 17 нФ на 305 метров (1000 футов) при частоте сигнала 1 кГц и температуре окружающей среды 20°С.

Для присоединения витых пар используются разъемы (коннекторы) типа RJ-45, похожие на разъемы, используемые в телефонах (RJ-11), но несколько большие по размеру. Разъемы RJ-45 имеют восемь контактов вместо четырех в случае RJ-11. Присоединяются разъемы к кабелю с помощью специальных обжимных инструментов. При этом золоченые игольчатые контакты разъема прокалывают изоляцию каждого провода, входят между его жилами и обеспечивают надежное и качественное соединение. Надо учитывать, что при установке разъемов стандартом

допускается расплетение витой пары кабеля на длину не более одного сантиметра.

Чаще всего витые пары используются для передачи данных в одном направлении (точка-точка), то есть в топологиях типа звезда или кольцо. Топология шина обычно ориентируется на коаксиальный кабель. Поэтому внешние терминаторы, согласующие неподключенные концы кабеля, для витых пар практически никогда не применяются.

Кабели выпускаются с двумя типами внешних оболочек:

- Кабель в поливинилхлоридной (ПВХ, PVC) оболочке дешевле и предназначен для работы в сравнительно комфортных условиях эксплуатации.
- Кабель в тефлоновой оболочке дороже и предназначен для более жестких условий эксплуатации.

Кабель в ПВХ оболочке называется еще non-plenum, а в тефлоновой – plenum. Термин plenum обозначает в данном случае пространство под фальшполом и над подвесным потолком, где удобно размещать кабели сети. Для прокладки в этих скрытых от глаз пространствах как раз удобнее кабель в тефлоновой оболочке, который, в частности, горит гораздо хуже, чем ПВХ – кабель, и не выделяет при этом ядовитых газов в большом количестве.

Еще один важный параметр любого кабеля, который жестко не определяется стандартом, но может существенно повлиять на работоспособность сети, – это скорость распространения сигнала в кабеле или, другими словами, задержка распространения сигнала в кабеле в расчете на единицу длины.

Стоит также отметить, что каждый из проводов, входящих в кабель на основе витых пар, как правило, имеет свой цвет изоляции, что существенно упрощает монтаж разъемов, особенно в том случае, когда концы кабеля находятся в разных комнатах, и контроль с помощью приборов затруднен.

Примером кабеля с экранированными витыми парами может служить кабель STP IBM типа 1, который включает в себя две экранированные витые

пары AWG типа 22. Волновое сопротивление каждой пары составляет 150 Ом. Для этого кабеля применяются специальные разъемы, отличающиеся от разъемов для неэкранированной витой пары (например, DB9). Имеются и экранированные версии разъема RJ-45.

Коаксиальные кабели

Коаксиальный кабель представляет собой электрический кабель, состоящий из центрального медного провода и металлической оплетки (экрана), разделенных между собой слоем диэлектрика (внутренней изоляции) и помещенных в общую внешнюю оболочку (рис. 3).

Рис. 2.3. Коаксиальный кабель

Коаксиальный кабель до недавнего времени был очень популярен, что связано с его высокой помехозащищенностью (благодаря металлической оплетке), более широкими, чем в случае витой пары, полосами пропускания (свыше 1ГГц), а также большими допустимыми расстояниями передачи (до километра). К нему труднее механически подключиться для несанкционированного прослушивания сети, он дает также заметно меньше электромагнитных излучений вовне. Однако монтаж и ремонт коаксиального кабеля существенно сложнее, чем витой пары, а стоимость его выше (он дороже примерно в 1,5 – 3 раза). Сложнее и установка разъемов на концах кабеля. Сейчас его применяют реже, чем витую пару. Стандарт EIA/TIA-568 включает в себя только один тип коаксиального кабеля, применяемый в сети Ethernet.

Основное применение коаксиальный кабель находит в сетях с топологией типа шина. При этом на концах кабеля обязательно должны устанавливаться терминаторы для предотвращения внутренних отражений сигнала, причем один (и только один!) из терминаторов должен быть заземлен. Без заземления металлическая оплетка не защищает сеть от

внешних электромагнитных помех и не снижает излучение передаваемой по сети информации во внешнюю среду. Но при заземлении оплетки в двух или более точках из строя может выйти не только сетевое оборудование, но и компьютеры, подключенные к сети. Терминаторы должны быть обязательно согласованы с кабелем, необходимо, чтобы их сопротивление равнялось волновому сопротивлению кабеля. Например, если используется 50-омный кабель, для него подходят только 50-омные терминаторы.

Реже коаксиальные кабели применяются в сетях с топологией звезда (например, пассивная звезда в сети Arcnet). В этом случае проблема согласования существенно упрощается, так как внешних терминаторов на свободных концах не требуется.

Волновое сопротивление кабеля указывается в сопроводительной документации. Чаще всего в локальных сетях применяются 50-омные (RG-58, RG-11, RG-8) и 93-омные кабели (RG-62). Распространенные в телевизионной технике 75-омные кабели в локальных сетях не используются. Марок коаксиального кабеля немного. Он не считается особо перспективным. Не случайно в сети Fast Ethernet не предусмотрено применение коаксиальных кабелей. Однако во многих случаях классическая шинная топология (а не пассивная звезда) очень удобна. Как уже отмечалось, она не требует применения дополнительных устройств – концентраторов.

Существует два основных типа коаксиального кабеля:

- тонкий (thin) кабель, имеющий диаметр около 0,5 см, более гибкий;
- толстый (thick) кабель, диаметром около 1 см, значительно более жесткий. Он представляет собой классический вариант коаксиального кабеля, который уже почти полностью вытеснен современным тонким кабелем.

Тонкий кабель используется для передачи на меньшие расстояния, чем толстый, поскольку сигнал в нем затухает сильнее. Зато с тонким кабелем гораздо удобнее работать: его можно оперативно проложить к каждому компьютеру, а толстый требует жесткой фиксации на стене помещения.

Подключение к тонкому кабелю (с помощью разъемов BNC байонетного типа) проще и не требует дополнительного оборудования. А для подключения к толстому кабелю надо использовать специальные довольно дорогие устройства, прокалывающие его оболочки и устанавливающие контакт как с центральной жилой, так и с экраном. Толстый кабель примерно вдвое дороже, чем тонкий, поэтому тонкий кабель применяется гораздо чаще.

Как и в случае витых пар, важным параметром коаксиального кабеля является тип его внешней оболочки. Точно так же в данном случае применяются как non-plenum (PVC), так и plenum кабели. Естественно, тефлоновый кабель дороже поливинилхлоридного. Обычно тип оболочки можно отличить по окраске (например, для PVC кабеля фирма Belden использует желтый цвет, а для тефлонового – оранжевый).

Типичные величины задержки распространения сигнала в коаксиальном кабеле составляют для тонкого кабеля около 5 нс/м, а для толстого – около 4,5 нс/м.

Существуют варианты коаксиального кабеля с двойным экраном (один экран расположен внутри другого и отделен от него дополнительным слоем изоляции). Такие кабели имеют лучшую помехозащищенность и защиту от прослушивания, но они немного дороже обычных.

В настоящее время считается, что коаксиальный кабель устарел, в большинстве случаев его вполне может заменить витая пара или оптоволоконный кабель. И новые стандарты на кабельные системы уже не включают его в перечень типов кабелей.

Оптоволоконные кабели

Оптоволоконный (он же волоконно-оптический) кабель – это принципиально иной тип кабеля по сравнению с рассмотренными двумя типами электрического или медного кабеля. Информация по нему передается не электрическим сигналом, а световым. Главный его элемент – это прозрачное стекловолокно, по которому свет проходит на огромные расстояния (до десятков километров) с незначительным ослаблением.

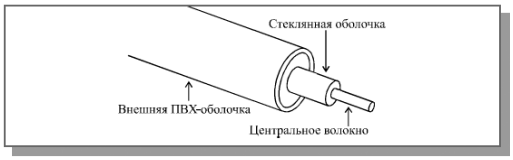


Рис. 4. Структура оптоволоконного кабеля

Структура оптоволоконного кабеля очень проста и похожа на структуру коаксиального электрического кабеля (рис. 4). Только вместо центрального медного провода здесь используется тонкое (диаметром около 1 – 10 мкм) стекловолокно, а вместо внутренней изоляции – стеклянная или пластиковая оболочка, не позволяющая свету выходить за пределы стекловолокна. В данном случае речь идет о режиме так называемого полного внутреннего отражения света от границы двух веществ с разными коэффициентами преломления (у стеклянной оболочки коэффициент преломления значительно ниже, чем у центрального волокна). Металлическая оплетка кабеля обычно отсутствует, так как экранирование от внешних электромагнитных помех здесь не требуется. Однако иногда ее все-таки применяют для механической защиты от окружающей среды (такой кабель иногда называют броневым, он может объединять под одной оболочкой несколько оптоволоконных кабелей).

Оптоволоконный кабель обладает исключительными характеристиками по помехозащищенности и секретности передаваемой информации. Никакие внешние электромагнитные помехи в принципе не способны исказить световой сигнал, а сам сигнал не порождает внешних электромагнитных излучений. Подключиться к этому типу кабеля для несанкционированного прослушивания сети практически невозможно, так как при этом нарушается целостность кабеля. Теоретически возможная полоса пропускания такого кабеля достигает величины 10^{12} Гц, то есть 1000 ГГц, что несравнимо выше, чем у электрических кабелей. Стоимость оптоволоконного кабеля постоянно снижается и сейчас примерно равна стоимости тонкого коаксиального кабеля.

Типичная величина затухания сигнала в оптоволоконных кабелях на частотах, используемых в локальных сетях, составляет от 5 до 20 дБ/км, что примерно соответствует показателям электрических кабелей на низких частотах. Но в случае оптоволоконного кабеля при росте частоты передаваемого сигнала затухание увеличивается очень незначительно, и на больших частотах (особенно свыше 200 МГц) его преимущества перед электрическим кабелем неоспоримы, у него просто нет конкурентов.

Однако оптоволоконный кабель имеет и некоторые недостатки.

Самый главный из них – высокая сложность монтажа (при установке разъемов необходима микронная точность, от точности скола стекловолокна и степени его полировки сильно зависит затухание в разъеме). Для установки разъемов применяют сварку или склеивание с помощью специального геля, имеющего такой же коэффициент преломления света, что и стекловолокно. В любом случае для этого нужна высокая квалификация персонала и специальные инструменты. Поэтому чаще всего оптоволоконный кабель продается в виде заранее нарезанных кусков разной длины, на обоих концах которых уже установлены разъемы нужного типа. Следует помнить, что некачественная установка разъема резко снижает допустимую длину кабеля, определяемую затуханием.

Также надо помнить, что использование оптоволоконного кабеля требует специальных оптических приемников и передатчиков, преобразующих световые сигналы в электрические и обратно, что порой существенно увеличивает стоимость сети в целом.

Оптоволоконные кабели допускают разветвление сигналов (для этого производятся специальные пассивные разветвители (couplers) на 2—8 каналов), но, как правило, их используют для передачи данных только в одном направлении между одним передатчиком и одним приемником. Ведь любое разветвление неизбежно сильно ослабляет световой сигнал, и если разветвлений будет много, то свет может просто не дойти до конца сети.

Кроме того, в разветвителе есть и внутренние потери, так что суммарная мощность сигнала на выходе меньше входной мощности.

Оптоволоконный кабель менее прочен и гибок, чем электрический. Типичная величина допустимого радиуса изгиба составляет около 10 – 20 см, при меньших радиусах изгиба центральное волокно может сломаться. Плохо переносит кабель и механическое растяжение, а также раздавливающие воздействия.

Чувствителен оптоволоконный кабель и к ионизирующим излучениям, из-за которых снижается прозрачность стекловолокна, то есть увеличивается затухание сигнала. Резкие перепады температуры также негативно сказываются на нем, стекловолокно может треснуть.

Применяют оптоволоконный кабель только в сетях с топологией звезда и кольцо. Никаких проблем согласования и заземления в данном случае не существует. Кабель обеспечивает идеальную гальваническую развязку компьютеров сети. В будущем этот тип кабеля, вероятно, вытеснит электрические кабели или, во всяком случае, сильно потеснит их. Запасы меди на планете истощаются, а сырья для производства стекла более чем достаточно.

Существуют два различных типа оптоволоконного кабеля:

- многомодовый или мультимодовый кабель, более дешевый, но менее качественный;
- одномодовый кабель, более дорогой, но имеет лучшие характеристики по сравнению с первым.

Суть различия между этими двумя типами сводится к разным режимам прохождения световых лучей в кабеле.

Рис. 5. Распространение света в одномодовом кабеле

В одномодовом кабеле практически все лучи проходят один и тот же путь, в результате чего они достигают приемника одновременно, и форма

сигнала почти не искажается (рис. 5). Одномодовый кабель имеет диаметр центрального волокна около 1,3 мкм и передает свет только с такой же длиной волны (1,3 мкм). Дисперсия и потери сигнала при этом очень незначительны, что позволяет передавать сигналы на значительно большее расстояние, чем в случае применения многомодового кабеля. Для одномодового кабеля применяются лазерные приемопередатчики, использующие свет исключительно с требуемой длиной волны. Такие приемопередатчики пока еще сравнительно дороги и не долговечны. Однако в перспективе одномодовый кабель должен стать основным типом благодаря своим прекрасным характеристикам. К тому же лазеры имеют большее быстродействие, чем обычные светодиоды. Затухание сигнала в одномодовом кабеле составляет около 5 дБ/км и может быть даже снижено до 1 дБ/км.

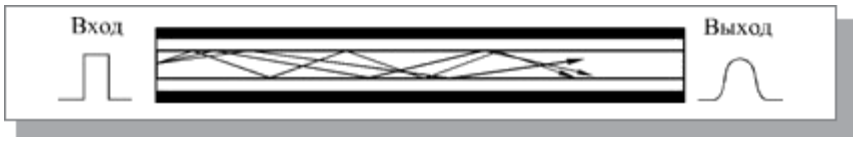


Рис. 6. Распространение света в многомодовом кабеле

В многомодовом кабеле траектории световых лучей имеют заметный разброс, в результате чего форма сигнала на приемном конце кабеля искажается (рис. 6). Центральное волокно имеет диаметр 62,5 мкм, а диаметр внешней оболочки 125 мкм (это иногда обозначается как 62,5/125). Для передачи используется обычный (не лазерный) светодиод, что снижает стоимость и увеличивает срок службы приемопередатчиков по сравнению с одномодовым кабелем. Длина волны света в многомодовом кабеле равна 0,85 мкм, при этом наблюдается разброс длин волн около 30 – 50 нм. Допустимая длина кабеля составляет 2 – 5 км. Многомодовый кабель – это основной тип оптоволоконного кабеля в настоящее время, так как он дешевле и доступнее. Затухание в многомодовом кабеле больше, чем в одномодовом и составляет 5 – 20 дБ/км.

Типичная величина задержки для наиболее распространенных кабелей составляет около 4—5 нс/м, что близко к величине задержки в электрических кабелях.

Оптоволоконные кабели, как и электрические, выпускаются в исполнении plenum и non-plenum.

Бескабельные каналы связи

Кроме кабельных каналов в компьютерных сетях иногда используются также бескабельные каналы. Их главное преимущество состоит в том, что не требуется никакой прокладки проводов (не надо делать отверстий в стенах, закреплять кабель в трубах и желобах, прокладывать его под фальшполами, над подвесными потолками или в вентиляционных шахтах, искать и устранять повреждения). К тому же компьютеры сети можно легко перемещать в пределах комнаты или здания, так как они ни к чему не привязаны.

Радиоканал использует передачу информации по радиоволнам, поэтому теоретически он может обеспечить связь на многие десятки, сотни и даже тысячи километров. Скорость передачи достигает десятков мегабит в секунду (здесь многое зависит от выбранной длины волны и способа кодирования).

Особенность радиоканала состоит в том, что сигнал свободно излучается в эфир, он не замкнут в кабель, поэтому возникают проблемы совместимости с другими источниками радиоволн (радио- и телевещательными станциями, радарами, радиолюбительскими и профессиональными передатчиками и т.д.). В радиоканале используется передача в узком диапазоне частот и модуляция информационным сигналом сигнала несущей частоты.

Главным недостатком радиоканала является его плохая защита от прослушивания, так как радиоволны распространяются неконтролируемо. Другой большой недостаток радиоканала – слабая помехозащищенность.

Для локальных беспроводных сетей (WLAN – Wireless LAN) в настоящее время применяются подключения по радиоканалу на небольших расстояниях (обычно до 100 метров) и в пределах прямой видимости. Чаще всего используются два частотных диапазона – 2,4 ГГц и 5 ГГц. Скорость передачи – до 54 Мбит/с. Распространен вариант со скоростью 11 Мбит/с.

Сети WLAN позволяют устанавливать беспроводные сетевые соединения на ограниченной территории (обычно внутри офисного или университетского здания или в таких общественных местах, как аэропорты). Они могут использоваться во временных офисах или в других местах, где прокладка кабелей неосуществима, а также в качестве дополнения к имеющейся проводной локальной сети, призванного обеспечить пользователям возможность работать перемещаясь по зданию.

Популярная технология Wi-Fi (Wireless Fidelity) позволяет организовать связь между компьютерами числом от 2 до 15 с помощью концентратора (называемого точка доступа, Access Point, AP), или нескольких концентраторов, если компьютеров от 10 до 50. Кроме того, эта технология дает возможность связать две локальные сети на расстоянии до 25 километров с помощью мощных беспроводных мостов. Для примера на рис. 7 показано объединение компьютеров с помощью одной точки доступа. Важно, что многие мобильные компьютеры (ноутбуки) уже имеют встроенный контроллер Wi-Fi, что существенно упрощает их подключение к беспроводной сети.

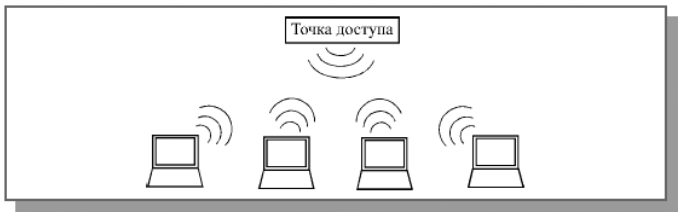


Рис. 7. Объединение компьютеров с помощью технологии Wi-Fi

Радиоканал широко применяется в глобальных сетях как для наземной, так и для спутниковой связи. В этом применении у радиоканала нет конкурентов, так как радиоволны могут дойти до любой точки земного шара.

Тема 4. Адресация в сетях. межсетевое взаимодействие.

Адрес, определяемый протоколом IP (Internetwork Protocol), состоит из четырех байтов, записываемых традиционно в десятичной системе счисления и разделяемых точкой. Адрес сетевого интерфейса eth0 из примера – 192.168.102.125. Второй сетевой интерфейс из примера, lo, – так называемая заглушка (loopback), которая используется для организации сетевых взаимодействий компьютера с самим собой: любой посланный в заглушку пакет немедленно обрабатывается как принятый оттуда. Заглушка обычно имеет адрес 127.0.0.1.

Отдельная среда передачи данных (локальная сеть) также имеет собственный адрес. Если представить IP-адрес в виде линейки из 32 битов, она строго разделяется на две части: столько-то битов слева отводится под адрес сети, а оставшиеся – под адрес абонента в этой сети. Для того чтобы определить размер адреса сети, используется сетевая маска – линейка из 32 битов, в которой на месте адреса сети стоят единицы, а на месте адреса компьютера – нули. При наложении маски на IP-адрес все единицы в нем, которым соответствуют нули в маске, превращаются в нули[2)]. Таким образом вычисляется IP-адрес сети. В примере сетевая маска интерфейса eth0 равна 255.255.255.0, т. е. 24 единицы и 8 нулей. Тогда IP-адрес сети будет равен 192.168.102.0. Мефодий заметил, что если сетевая маска выровнена по границе байта, производить двоичные операции вообще не надо: так, в примере можно было просто сказать, что адрес сети занимает три байта, а адрес абонента – оставшийся один.

Заметим, что адрес сети может содержать значащие нули: например, в адресе 10.0.0.1 при сетевой маске 255.255.0.0 адрес сети занимает два байта, из которых второй – полностью нулевой. Чтобы не гадать, какие нули – значащие, а какие – отрезаны маской, к адресу сети принято приписывать уточнение вида /количество_единиц_в_маске. В приведенном случае адрес сети выглядел бы так: 10.0.0.0/16, а в предыдущем – 192.168.102.0/24.

IP-адрес, составленный из адреса сети, за которым следуют все единицы (в примере – 192.168.102.255), называется широковещательный адрес: любой принадлежащий сети 192.168.102.0 компьютер, получивший IP-пакет с адресом получателя 192.168.102.255, должен обработать его, как если бы в поле "получатель" стоял его собственный IP-адрес.

Когда компьютер с некоторым IP-адресом решает отправить пакет другому компьютеру, он выясняет, принадлежит ли адресат той же локальной сети, что и отправитель (т. е. подключены ли они к одной среде передачи данных). Делается это так: на IP-адрес получателя накладывается сетевая маска, и таким образом вычисляется адрес сети, которой принадлежит получатель. Если этот адрес совпадает с адресом сети отправителя, значит, оба находятся в одной локальной сети. Это, в свою очередь, означает, что аппаратный адрес (MAC) получателя должен быть отправителю известен.

В TCP/IP вопрос о том, как обеспечить нескольким абонентам сети возможность передавать данные, не мешая друг другу, решен с помощью разделения пакетов данных. разделение пакетов предполагает, что данные передаются не единым блоком, а по частям, пакетами. Алгоритмы, определяющие, когда абоненту разрешено посылать следующий пакет, могут быть разными, но результат всегда один: в сети передаются попеременно фрагменты всех сеансов передачи данных. В сильно загруженном состоянии такая сеть может просто не принять очередной пакет от абонента-отправителя, и тому придется ждать удобного случая, чтобы все-таки "пропихнуть" его в переполненную другими пакетами среду. Таким образом, обеспечить гарантированное время передачи одного пакета в сетях с разделением пакетов бывает довольно сложно, хотя существуют алгоритмы, позволяющие это сделать.

Противоположность метода разделения пакетов – метод разделения каналов, который предполагает, что в сети имеется определенное число каналов передачи данных, которые абоненты сети арендуют на все время

передачи. По такому принципу построены, например, телефонные линии: дозвонившись, мы арендуем канал связи между двумя телефонными аппаратами, и до тех пор, пока этот канал занят, невозможно ни воспользоваться им кому-то другому, ни организовать параллельно передачу данных откуда-нибудь еще. Главное достоинство сетей с разделеним каналов – постоянная (за вычетом помех на линии) скорость передачи данных. Основной недостаток – ограниченное количество каналов передачи. Проектировать среду передачи так, чтобы каждый абонент был связан с каждым отдельным каналом, имеет смысл только тогда, когда абонентов очень мало: количество каналов будет пропорционально квадрату количества абонентов. Количество каналов в большой сети будет существенно меньшим, и ровно столько сеансов передачи данных можно будет в этой сети установить. Попытка соединиться с абонентом, когда все каналы уже заняты, окончится неудачей. Если вернуться к сети с разделением пакетов, то можно заметить, что на каждом уровне под пакетом понимается разное. С точки зрения интерфейсного уровня пакет – это ограниченный возможностями среды передачи данных фрагмент, в котором необходимо дополнительно указать, какое устройство из числа подключенных к среде передачи данных его отправило и какому устройству он предназначен. С точки зрения сетевого уровня размер пакета определяется удобством его обработки, а дополнительно в нем надо указать уникальные для всей сети адреса отправителя и получателя (а также тип протокола и многое другое). С точки зрения транспортного уровня размер пакета определяется качеством связи (чем меньше пакет, тем ниже вероятность порчи, но тем больше теряется на дополнительной информации: идентификатор сеанса, тип, специальные поля, описывающие логику связи и т.п.). Наконец, если на прикладном уровне определено понятие "пакет", то его размер и содержимое определяются протоколом прикладного уровня.

Таким образом, процесс передачи данных выглядит так: порция данных прикладного уровня нарезается на части, соответствующие размеру пакета

транспортного уровня (фрагментируется), к каждому фрагменту приписывается транспортная служебная информация, и получаются пакеты транспортного уровня. Каждый пакет транспортного уровня может быть опять-таки фрагментирован для передачи по сети, к каждому получившемуся фрагменту добавляется служебная информация сетевого уровня, что дает последовательность сетевых пакетов. Каждый из сетевых пакетов тоже может быть фрагментирован до размера, "пролезающего" через конкретное сетевое устройство, – из него формируются пакеты интерфейсного уровня (фреймы). Наконец, к каждому фрейму само устройство (по крайне мере, так это сделано в Ethernet) приписывает некоторый ключ, по которому принимающее устройство распознает начало фрейма. В таком виде данные передадутся по проводам. Процесс "заворачивания" пакетов более высокого уровня в пакеты более низкого уровня называется инкапсуляцией.

Компьютер, получивший фрейм, выполняет процедуры, обратные инкапсуляции и фрагментации: пакеты низкого уровня освобождаются от служебной информации и накапливаются до тех пор, пока не сформируется пакет более высокого уровня. Затем этот пакет отсылается на уровень выше и все повторяется до тех пор, пока освобожденные от всей дополнительной информации и заново собранные воедино данные не попадут к пользователю (или к программе, которая их обрабатывает).

Сетевой пакет. Единица передачи информации в компьютерной сети. Помимо передаваемых данных содержит служебную информацию, в частности, идентификаторы отправителя и адресата, контрольную сумму, поля используемого протокола. Наибольший размер пакета определяется чаще всего не объемом передаваемых данных, а требованиями протокола и необходимостью разделять сеть передачи данных между несколькими абонентами.

Аппаратный и интерфейсный уровни

Итак, на аппаратном уровне возможна какая угодно среда передачи данных, сеть начинается в месте подключения к этой среде, то есть на

сетевом интерфейсе. Список сетевых интерфейсов и их настроек в системе можно посмотреть с помощью команды ifconfig (от interface configuration):

Утилитой ifconfig пользуется, в основном, сама система или администратор; некоторые данные ifconfig получает, обращаясь с системным вызовом ioctl() к открытому сетевому сокету, а некоторые считывает из /proc. Название сетевого интерфейса состоит из его типа и порядкового номера (каким по счету его распознало ядро). Все сетевые интерфейсы Ethernet в Linux называются ethномер, начиная с eth0. Параметр MTU (Maximum Transfer Unit) определяет наибольший размер фрейма.

Большинство других параметров относятся к сетевому уровню, но как минимум еще один – HWaddr – относится к уровню интерфейсному.

Сетевой интерфейс. Точка взаимодействия утилит Linux с реализацией TCP/IP в ядре системы. Как правило, имеет уникальный сетевой адрес. Интерфейсу может соответствовать некоторое сетевое оборудование (например, карта Ethernet), в этом случае определен также и его интерфейсный адрес.

HWaddr (от HardWare address, аппаратный адрес) – это уникальный внутри среды передачи данных идентификатор сетевого устройства. В Ethernet аппаратный адрес называется MAC-address (от Media Access Control, управление доступом к среде), он состоит из шести байтов, которые принято записывать в шестнадцатиричной системе исчисления и разделять двоеточиями. Каждая Ethernet-карта имеет собственный уникальный MAC-address (в примере – 00:0C:29:56:C1:36), поэтому его легко использовать для определения отправителя и получателя в рамках одной Ethernet-среды. Если идентификатор получателя неизвестен, используется аппаратный широковещательный адрес, FF:FF:FF:FF:FF:FF. Сетевая карта, получив широковещательный фрейм или фрейм, MAC-адрес получателя в котором совпадает с ее MAC-адресом, обязана отправить его на обработку системе.

Термин "Media Access Control" имеет отношение к алгоритму, с помощью которого решается задача очередности передачи. Алгоритм базируется на трех принципах:

1. Прослушивание среды. Каждое устройство умеет определять, идет ли в данное время передача данных по среде. Если среда свободна, устройство имеет право само передавать данные.
2. Обнаружение коллизий. Если решение о начале передачи данных одновременно приняли несколько устройств, в среде возникнет коллизия, и распознать, где чьи были данные, становится невозможно. Зато устройства всегда замечают произошедшую коллизию, и передают данные повторно.
3. Случайное время ожидания перед повтором. Если бы после коллизии все устройства начали одновременно повторять передачу данных, случилась бы новая коллизия. Поэтому каждое устройство выжидает некоторое случайное время, и только после этого повторяет передачу. Если повторная коллизия все-таки возникает, устройство ждет вдвое дольше[1)]. так происходит до тех пор, пока не будет превышено допустимое время ожидания, после чего системе сообщается об ошибке.

Приведенный алгоритм имеет два недостатка. Во-первых, уже на интерфейсном уровне время передачи одного пакета может быть любым, так как неопределенное промедление с передачей предусмотрено протоколом. Во-вторых, сеть Ethernet считается хорошо загруженной, если на протяжении некоторого промежутка времени в среднем треть этого времени было потрачена на передачу данных, а две трети времени среда была свободна. Сеть Ethernet, нагруженная наполовину, работает очень медленно и с большим числом коллизий, а сеть, нагруженная на две трети, считается неработающей. Это – плата за отсутствие синхронизации работы всех устройств в сети.

Сетевой уровень

Создатели первых сетей, объединяющих несколько сред передачи данных, для идентификации абонента таких сетей пытались использовать те же аппаратные адреса. Это оказалось делом неблагодарным: если в Ethernet аппаратный адрес уникален всегда, то в других сетях аппаратные адреса могут быть уникальны только в рамках одной среды (например, все устройства нумеруются, начиная с 0) или даже могут выдаваться динамически, да и форматы аппаратных адресов в разных средах различны. Возникла необходимость присвоить каждому сетевому интерфейсу некоторый единственный на всю глобальную сеть адрес, который бы не зависел от среды передачи данных и всегда имел один и тот же формат.

Маршрутизация

Более сложный вопрос встает, если IP-адрес компьютера-адресата не входит в локальную сеть компьютера-отправителя. Ведь и в этом случае пакет необходимо отослать какому-то абоненту локальной сети, с тем, чтобы тот перенаправил его дальше. Этот абонент, маршрутизатор, подключен к нескольким сетям, и ему вменяется в обязанность пересылать пакеты между ними по определенным правилам. В самом простом случае таких сетей две: "внутренняя", к которой подключены компьютеры, и "внешняя", соединяющая маршрутизатор со всей глобальной сетью. Таблицу, управляющую маршрутизацией пакетов, можно просмотреть с помощью команды netstat -r или route (обе команды имеют ключ "-n", заставляющий их использовать в выдаче IP-адреса, а не имена компьютеров):

Транспортный уровень

Транспортных протоколов в TCP/IP два – это TCP (Transmission Control Protocol, протокол управления соединением) и UDP (User Datagram Protocol). UDP устроен просто. Пользовательские данные помещаются в единственный транспортный пакет-датаграмму, которой приписываются обычные для транспортного уровня данные: адреса и порты отправителя и получателя, после чего пакет уходит в сеть искать адресата. Проверять, был ли адресат

способен этот пакет принять, дошел ли пакет до него и не испортился ли по дороге, предоставляется следующему – прикладному – уровню.

Иное дело – TCP. Этот протокол очень заботится о том, чтобы передаваемые данные дошли до адресата в целости и сохранности. Для этого предпринимаются следующие действия:

1. Устанавливается соединение

Перед тем, как начать передавать данные, TCP проверяет, способен ли адресат их принимать. Если адресат отвечает согласием на открытие соединения, устанавливается двусторонняя связь между ним и отправителем. Помимо адресов отправителя и адресата и номеров портов на отправителе и адресате, в TCP-соединении участвуют два номера последовательности (SEQuential Number, SEQN), с помощью которых каждая сторона проверяет, не потерялись ли пакеты по пути, не перепутались ли.

2. Обрабатываются подтверждения

Двусторонняя связь нужна еще и потому, что на каждый TCP-пакет с любой стороны требуется подтверждение того, что этот пакет принят. Упрощенно можно представить дело так, что отправитель и адресат по очереди обмениваются пакетами, каждый из которых содержит подтверждение только что принятого, и, возможно, полезные данные. Если происходит какая-то ошибка, она возвращается вместо подтверждения и отправитель обрабатывает ее (например, посылает пакет еще раз).

3. Отслеживаются состояния абонентов

С первым же подтверждением каждый из абонентов передает размер т. н. скользящего окна (sliding window). Этот размер показывает, сколько еще данных готов принять адресат. Отправитель посылает сразу несколько пакетов суммарным размером с это окно, а после ждет подтверждения об их принятии. Когда приходит подтверждение первого из пакетов в окне, окно "скользит" вперед: теперь оно начинается со второго пакета, и в него

попадает один или несколько еще не посланных пакетов. Если адресат может принять больше данных, он сообщает о большем размере окна, а если данные перерабатываться не успевают – о меньшем.

Кажется, что TCP – протокол во всех отношениях более удобный, чем UDP. Однако в тех случаях, когда пользовательские данные всегда помещаются в один пакет, зато самих пакетов идет очень много, посылать всего одну датаграмму намного выгоднее, чем всякий раз устанавливать соединение, пересылать данные и закрывать соединение (что требует, как минимум, по три пакета в каждую сторону). Очень трудно использовать TCP для широковещательных передач, когда число абонентов-адресатов весьма велико или вовсе неизвестно. Посмотреть параметры всех передаваемых через сетевой интерфейс пакетов можно с помощью команды tcpdump -pi интерфейс, хотя Мефодию не хватило поверхностного знания TCP/IP для того, чтобы понять выдачу этой команды.

Прикладной уровень

Как бы ни был надежен протокол TCP, он не имеет никакого понятия о том, что же, собственно, за данные с его помощью передаются. Да и не должен: принцип разделения уровней не позволяет заглядывать "внутрь" передаваемого пакета, и способов наверняка распознать используемый в нем прикладной протокол нет. Прикладной уровень, в отличие от транспортного, предусматривает сколько угодно протоколов передачи данных. Интерпретация данных, в конце концов, дело уже не ядра, а какой-нибудь программы ("приложения", как правило, демона). Для того чтобы можно было предположить, какой протокол используется при передаче данных, а также для того, чтобы система могла передать эти данные соответствующей программе, еще на транспортном уровне было введено понятие порта.

Клиент-серверная модель

С точки зрения прикладного уровня, порт – это идентификатор сервиса, предоставляемого системой. В самом деле, практически любой акт передачи данных выглядит, как если бы некий клиент, которому эти данные нужны,

запрашивал их у сервера, который может их предоставить[1)]. Отношения между программами, которые связываются по сети друг с другом, почти всегда асимметричны: одной что-то надо, у другой это что-то есть. При установлении соединения и приложение (программа-клиент), и служба (программа-сервер) используют механизм сокетов, описанный в лекции 11, однако ведут себя по-разному.

Служба, запускаясь на сервере, создает сетевой сокет и прикрепляет его к определенному порту сервера с помощью системного вызова bind(). Затем она регистрируется в качестве обработчика запросов (listener), приходящих на этот порт. Служба ждет запросов, и когда они поступают, предпринимает какие-нибудь действия, например, считывает пришедшие данные и анализирует их в соответствии со своим протоколом, отсылает какие-то данные абоненту, пославшему запрос и т. п.

Приложение, запускаясь на клиенте, также создает сокет и присоединяется с его помощью к тому же порту на сервере, где запущена служба, используя системный вызов connect(). Затем оно, как и служба, посылает и получает данные. Разницы между обменом данными по сетевому сокету и по сокету в файловой системе нет. Очередность обмена данными определяется прикладным протоколом.

Тема 6. Безопасность и защита данных

Существует класс программ, которые были изначально написаны с целью уничтожения данных на чужом компьютере, похищения чужой информации, несанкционированного использования чужих ресурсов и т. п., или же приобрели такие свойства вследствие каких-либо причин. Такие программы несут вредоносную нагрузку и соответственно называются вредоносными.

Вирусы

Основная черта компьютерного вируса - это способность к саморазмножению.

Компьютерный вирус- это программа, способная создавать свои дубликаты (не обязательно совпадающие с оригиналом) и внедрять их в вычислительные сети и/или файлы, системные области компьютера и прочие выполняемые объекты. При этом дубликаты сохраняют способность к дальнейшему распространению.

Условно жизненный цикл любого компьютерного вируса можно разделить на пять стадий:

1. Проникновение на чужой компьютер
2. Активация
3. Поиск объектов для заражения
4. Подготовка копий
5. Внедрение копий

Путями проникновения вируса могут служить как мобильные носители, так и сетевые соединения - фактически, все каналы, по которым можно скопировать файл. Однако в отличие от червей, вирусы не используют сетевые ресурсы - заражение вирусом возможно, только если пользователь сам каким-либо образом его активировал. Например, скопировал или получил по почте зараженный файл и сам его запустил или просто открыл
После проникновения следует активация вируса. Это может происходить несколькими путями и в соответствии с выбранным методом вирусы делятся на такие виды:

- Загрузочные вирусы заражают загрузочные сектора жестких дисков и мобильных носителей.
- Файловые вирусы - заражают файлы. Отдельно по типу среды обитания в этой группе также выделяют:
    - Классические файловые вирусы - они различными способами внедряются в исполняемые файлы (внедряют свой вредоносный код или полностью их перезаписывают), создают файлы-двойники, свои копии в различных каталогах жесткого диска или используют особенности организации файловой системы

- Макровирусы, которые написаны на внутреннем языке, так называемых макросах какого-либо приложения. Подавляющее большинство макровирусов используют макросы текстового редактора Microsoft Word
- Скрипт-вирусы, написанные в виде скриптов для определенной командной оболочки - например, bat-файлы для DOS или VBS и JS - скрипты для Windows Scripting Host (WSH)

Дополнительным отличием вирусов от других вредоносных программ служит их жесткая привязанность к операционной системе или программной оболочке, для которой каждый конкретный вирус был написан. Это означает, что вирус для Microsoft Windows не будет работать и заражать файлы на компьютере с другой установленной операционной системой, например Unix. Точно также макровирус для Microsoft Word 2003 скорее всего не будет работать в приложении Microsoft Excel 97.

При подготовке своих вирусных копий для маскировки от антивирусов могут применять такие технологии как:

- Шифрование - в этом случае вирус состоит из двух частей: сам вирус и шифратор.
- Метаморфизм - при применении этого метода вирусные копии создаются путем замены некоторых команд на аналогичные, перестановки местами частей кода, вставки между ними дополнительного, обычно ничего не делающих команд.

Соответственно в зависимости от используемых методов вирусы можно делить на шифрованные, метаморфные и полиморфные, использующие комбинацию двух типов маскировки.

Основные цели любого компьютерного вируса - это распространение на другие ресурсы компьютера и выполнение специальных действий при определенных событиях или действиях пользователя (например, 26 числа каждого четного месяца или при перезагрузке компьютера). Специальные действия нередко оказываются вредоносными.

Черви

В отличие от вирусов черви - это вполне самостоятельные программы. Главной их особенностью также является способность к саморазмножению, однако при этом они способны к самостоятельному распространению с использованием сетевых каналов. Для подчеркивания этого свойства иногда используют термин "сетевой червь".

Червь (сетевой червь) - это вредоносная программа, распространяющаяся по сетевым каналам и способная к самостоятельному преодолению систем защиты компьютерных сетей, а также к созданию и дальнейшему распространению своих копий, не обязательно совпадающих с оригиналом.

Жизненный цикл червей состоит из таких стадий:

1. Проникновение в систему
2. Активация
3. Поиск объектов для заражения
4. Подготовка копий
5. Распространение копий

В зависимости от способа проникновения в систему черви делятся на типы:

- Сетевые черви используют для распространения локальные сети и Интернет
- Почтовые черви - распространяются с помощью почтовых программ
- IM-черви используют системы мгновенного обмена сообщениями
- IRC-черви распространяются по каналам IRC
- P2P-черви - при помощи пиринговых файлообменных сетей

После проникновения на компьютер, червь должен активироваться - иными словами запуститься. По методу активации все черви можно

разделить на две большие группы - на тех, которые требуют активного участия пользователя и тех, кто его не требует. На практике это означает, что бывают черви, которым необходимо, чтобы владелец компьютера обратил на них внимание и запустил зараженный файл, но встречаются и такие, которые делают это сами, например, используя ошибки в настройке или бреши в системе безопасности операционной системы. Отличительная особенность червей из первой группы - это использование обманных методов. Это проявляется, например, когда получатель инфицированного файла вводится в заблуждение текстом письма и добровольно открывает вложение с почтовым червем, тем самым его активируя. В последнее время наметилась тенденция к совмещению этих двух технологий - такие черви наиболее опасны и часто вызывают глобальные эпидемии.

Сетевые черви могут кооперироваться с вирусами - такая пара способна самостоятельно распространяться по сети (благодаря червю) и в то же время заражать ресурсы компьютера (функции вируса).

Трояны

Трояны или программы класса троянский конь, в отличие от вирусов и червей, не обязаны уметь размножаться. Это программы, написанные только с одной целью - нанести ущерб целевому компьютеру путем выполнения несанкционированных пользователем действий: кражи, порчи или удаления конфиденциальных данных, нарушения работоспособности компьютера или использования его ресурсов в неблаговидных целях.

Троян (троянский конь) - программа, основной целью которой является вредоносное воздействие по отношению к компьютерной системе.

Некоторые трояны способны к самостоятельному преодолению систем защиты компьютерной системы, с целью проникновения в нее. Однако в большинстве случаев они проникают на компьютеры вместе с вирусом либо червем - то есть такие трояны можно рассматривать как дополнительную вредоносную нагрузку, но не как самостоятельную программу. Нередко пользователи сами загружают троянские программы из Интернет.

Следовательно, жизненный цикл троянов состоит всего из трех стадий:

1. Проникновение в систему
2. Активация
3. Выполнение вредоносных действий

Как уже говорилось выше, проникать в систему трояны могут двумя путями - самостоятельно и в кооперации с вирусом или сетевым червем. В первом случае обычно используется маскировка, когда троян выдает себя за полезное приложение, которое пользователь самостоятельно копирует себе на диск (например, загружает из Интернет) и запускает. При этом программа действительно может быть полезна, однако наряду с основными функциями она может выполнять действия, свойственные трояну.

После проникновения на компьютер, трояну необходима активация и здесь он похож на червя - либо требует активных действий от пользователя или же через уязвимости в программном обеспечении самостоятельно заражает систему.

Поскольку главная цель написания троянов - это производство несанкционированных действий, они классифицируются по типу вредоносной нагрузки:

- Клавиатурные шпионы, постоянно находясь в оперативной памяти, записывают все данные, поступающие от клавиатуры с целью последующей их передачи своему автору.
- Похитители паролей предназначены для кражи паролей путем поиска на зараженном компьютере специальных файлов, которые их содержат.
- Утилиты скрытого удаленного управления - это трояны, которые обеспечивают несанкционированный удаленный контроль над инфицированным компьютером. Перечень действий, которые позволяет выполнять тот или иной троян, определяется его функциональностью, заложенной автором. Обычно это возможность скрыто загружать, отсылать, запускать или уничтожать файлы. Такие трояны могут быть

использованы как для получения конфиденциальной информации, так и для запуска вирусов, уничтожения данных.

- Анонимные SMTP-сервера и прокси-сервера - такие трояны на зараженном компьютере организовывают несанкционированную отправку электронной почты, что часто используется для рассылки спама.
- Утилиты дозвона в скрытом от пользователя режиме инициируют подключение к платным сервисам Интернет.
- Модификаторы настроек браузера меняют стартовую страницу в браузере, страницу поиска или еще какие-либо настройки, открывают дополнительные окна, имитируют нажатия на рекламные баннеры и т. п.
- Логические бомбы характеризуются способностью при срабатывании заложенных в них условий (в конкретный день, время суток, определенное действие пользователя или команды извне) выполнять какое-либо действие, например, удаление файлов.

Отдельно отметим, что существуют программы из класса троянов, которые наносят вред другим, удаленным компьютерам и сетям, при этом не нарушая работоспособности инфицированного компьютера. Яркие представители этой группы - организаторы DDoS-атак.

Другие вредоносные программы

Кроме вирусов, червей и троянов существует еще множество других вредоносных программ, для которых нельзя привести общий критерий. Однако среди них можно выделить небольшие группы. Это в первую очередь:

- Условно опасные программы, то есть такие, о которых нельзя однозначно сказать, что они вредоносны. Такие программы обычно становятся опасными только при определенных условиях или действиях пользователя. К ним относятся:
  - Riskware - вполне легальные программы, которые сами по себе не опасны, но обладают функционалом, позволяющим злоумышленнику использовать их с вредоносными целями. К riskware относятся

обычные утилиты удаленного управления, которыми часто пользуются администраторы больших сетей, клиенты IRC, программы для загрузки файлов из Интернет, утилиты восстановления забытых паролей и другие.

- Рекламные утилиты (adware) - условно-бесплатные программы, которые в качестве платы за свое использование демонстрируют пользователю рекламу, чаще всего в виде графических баннеров. После официальной оплаты и регистрации обычно показ рекламы заканчивается и программы начинают работать в обычном режиме. Проблема adware кроется в механизмах, которые используются для загрузки рекламы на компьютер. Кроме того, что для этих целей часто используются программы сторонних и не всегда проверенных производителей, даже после регистрации такие модули могут автоматически не удаляться и продолжать свою работу в скрытом режиме. Однако среди adware-программ есть и вполне заслуживающие доверия - например, клиент ICQ.
- Pornware - к этому классу относятся утилиты, так или иначе связанные с показом пользователям информации порнографического характера. На сегодняшний день это программы, которые самостоятельно дозваниваются до порнографических телефонных служб, загружают из Интернет порнографические материалы или утилиты, предлагающие услуги по поиску и показу такой информации. Отметим, что к вредоносным программам относятся только те утилиты класса pornware, которые устанавливаются на компьютер пользователя несанкционированно - через уязвимость в операционной системы или браузера или при помощи троянов. Обычно это делается с целью насильственного показа рекламы платных порнографических сайтов или служб.

- Хакерские утилиты - К этому виду программ относятся программы скрытия кода зараженных файлов от антивирусной проверки

(шифровальщики файлов), автоматизации создания сетевых червей, компьютерных вирусов и троянских программ (конструкторы вирусов), наборы программ, которые используют хакеры для скрытного взятия под контроль взломанной системы (RootKit) и другие подобные утилиты. То есть такие специфические программы, которые обычно используют только хакеры.

- Злые шутки - программы, которые намеренно вводят пользователя в заблуждение путем показа уведомлений о, например, форматировании диска или обнаружении вирусов, хотя на самом деле ничего не происходит. Текст таких сообщений целиком и полностью отражает фантазию автора.

Политика безопасности

На домашнем компьютере пользователь сам устанавливает себе правила, которым он считает нужным следовать. По мере накопления знаний о работе компьютера и о вредоносных программах, он может сознательно менять настройки защиты или принимать решение об опасности тех или иных файлов и программ.

В большой организации все сложнее. Когда коллектив объединяет большое количество сотрудников, выполняющих разные функции и имеющих разную специализацию, сложно ожидать от всех разумного поведения с точки зрения безопасности. Поэтому в каждой организации правила работы с компьютером должны быть общими для всех сотрудников и утверждены официально. Обычно, документ, содержащий эти правила называется инструкцией пользователя. Кроме основных правил, перечисленных выше, он должен обязательно включать информацию о том, куда должен обращаться пользователь при возникновении ситуации, требующей вмешательства специалиста.

При этом инструкция пользователя в большинстве случаев содержит только правила, ограничивающие его действия. Правила использования программ в инструкцию могут входить только в самом ограниченном виде.

Поскольку большинство пользователей недостаточно компетентны в вопросах безопасности, они не должны, а часто и не могут менять настройки средств защиты и как-то влиять на их работу.

Но если не пользователи, то кто-то другой все-таки должен отвечать за настройку средств защиты и за управление ими. Обычно это специально назначенный сотрудник или группа сотрудников, которые сосредоточены на выполнении одной задачи - обеспечении безопасной работы сети.

Сотрудникам, ответственным за безопасность, приходится устанавливать и настраивать защитные программы на большом количестве компьютеров. Если на каждом компьютере заново решать, какие настройки безопасности должны быть установлены, несложно предположить, что разные сотрудники в разное время и на разных компьютерах установят пусть и похожие, но несколько разные настройки. В такой ситуации будет очень сложно оценить, насколько защищена организация в целом, т. к. никто не знает всех установленных параметров защиты.

Брандмауэры

Для того чтобы удаленно воспользоваться уязвимостью в программном обеспечении или операционной системе, нужно установить соединение и передать специально сформированный пакет данных. Следовательно можно защититься от таких попыток проникновения и заражения, путем запрета определенных соединений. Задачу контроля соединений успешно решают программы-брандмауэры.

Брандмауэр - это программа, которая следит за сетевыми соединениями и принимает решение о разрешении или запрещении новых соединений на основании заданного набора правил.

Правило брандмауэра как правило задается несколькими атрибутами:

- Приложение - определяет программу к которой относится правило, так что одни и те же действия могут быть разрешены одним программам и запрещены другим. Например, получать и отправлять почту разумно разрешить только программе - почтовому клиенту

- Протокол - определяет протокол, используемый для передачи данных. Обычно можно выбрать между двумя протоколами TCP и UDP [1)]
- Адреса - определяет, для соединений с каких адресов или на какие адреса будет действовать правило
- Порт - задает номера портов, на которые распространяется правило
- Направление - позволяет отдельно контролировать входящие и исходящие соединения
- Действие - определяет реакцию на обнаружение соединения, соответствующего остальным параметрам. Реакция может быть - разрешить, запретить или спросить у пользователя

Не обязательно давать конкретные значения всем атрибутам правила. Можно создать правило, которое будет запрещать входящие соединения на TCP порт 111 для всех приложений, или разрешать любые исходящие соединения для программы Internet Explorer.

Для борьбы с вирусами брандмауэры могут применяться в двояком качестве. Во-первых, брандмауэр можно успешно использовать для защиты от вредоносных программ, которые распространяются непосредственно по сети, используя уязвимости в операционной системе. Например, червь Sasser атакует службу Windows LSASS через TCP порт 445. Значит для защиты от червя достаточно создать в брандмауэре правило, запрещающее входящие соединения на этот порт. Если речь идет о домашнем компьютере, который использует сеть только для выхода в Интернет, такое способ защиты не будет иметь побочных эффектов. В организации с локальной сетью, где порт 445 используется для работы Windows-сети, могут возникнуть неудобства.
Брандмауэр можно использовать и для защиты от атак неизвестных вирусов. В случае домашнего компьютера, использующего сеть только для доступа в Интернет, можно запретить вообще все входящие соединения, и тем самым обезопасить себя от любых атак извне.

Второй аспект применения брандмауэров для защиты от вредоносных программ состоит в контроле исходящих соединений. Многие троянские

программы, да и черви, после выполнения вредоносной функции стремиться подать сигнал автору вируса. Например, троянская программа, ворующая пароли, будет пытаться переслать все найденные пароли на определенный сайт или почтовый адрес. Для того чтобы воспрепятствовать этому, можно настроить брандмауэр на блокирование всех неизвестных соединений: разрешить только соединения от доверенных программ, таких как используемый браузер, почтовый клиент, программа мгновенного обмена сообщений, а все остальные соединения запретить. В таком случае, вредоносная программа, даже попав на компьютер незамеченной, не сможет причинить реального ущерба.

Некоторые вредоносные программы не пытаются активно пересылать данные, а пассивно ожидают соединения на каком-то из портов. Если входящие соединения разрешены, то автор вредоносной программы сможет через некоторое время обратиться на этот порт и забрать нужную ему информацию или же передать вредоносной программе новые команды. Чтобы этого не произошло, брандмауэр должен быть настроен на запрет входящих соединений либо на все порты вообще, либо на все, кроме фиксированного перечня портов, используемых известными программами или операционной системой.

Межсетевые экраны прикладного уровня

Межсетевые экраны прикладного уровня, или прокси-экраны, представляют собой программные пакеты, базирующиеся на операционных системах общего назначения (таких как Windows NT и Unix) или на аппаратной платформе межсетевых экранов. Межсетевой экран обладает несколькими интерфейсами, по одному на каждую из сетей, к которым он подключен. Набор правил политики определяет, каким образом трафик передается из одной сети в другую. Если в правиле отсутствует явное разрешение на пропуск трафика, межсетевой экран отклоняет или аннулирует пакеты.

Правила политики безопасности усиливаются посредством использования модулей доступа. В межсетевом экране прикладного уровня каждому разрешаемому протоколу должен соответствовать свой собственный модуль доступа. Лучшими модулями доступа считаются те, которые построены специально для разрешаемого протокола. Например, модуль доступа FTP предназначен для протокола FTP и может определять, соответствует ли проходящий трафик этому протоколу и разрешен ли этот трафик правилами политики безопасности.

При использовании межсетевого экрана прикладного уровня все соединения проходят через него (см. рис. 1). Как показано на рисунке, соединение начинается на системе-клиенте и поступает на внутренний интерфейс межсетевого экрана. Межсетевой экран принимает соединение, анализирует содержимое пакета и используемый протокол и определяет, соответствует ли данный трафик правилам политики безопасности. Если это так, то межсетевой экран инициирует новое соединение между своим внешним интерфейсом и системой-сервером.

Межсетевые экраны прикладного уровня используют модули доступа для входящих подключений. Модуль доступа в межсетевом экране принимает входящее подключение и обрабатывает команды перед отправкой трафика получателю. Таким образом, межсетевой экран защищает системы от атак, выполняемых посредством приложений.

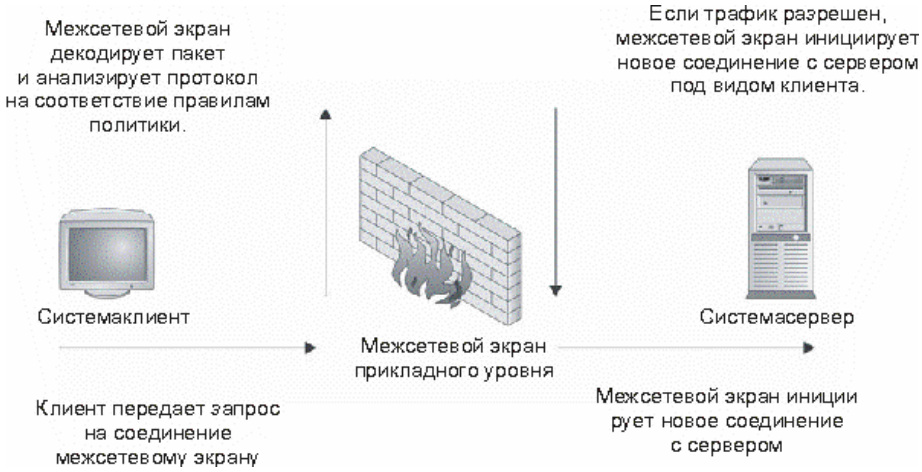


Рис. 1. Соединения модуля доступа межсетевого экрана прикладного уровня

Межсетевые экраны с пакетной фильтрацией

Межсетевые экраны с пакетной фильтрацией могут также быть программными пакетами, базирующимися на операционных системах общего назначения (таких как Windows NT и Unix) либо на аппаратных платформах межсетевых экранов. Межсетевой экран имеет несколько интерфейсов, по одному на каждую из сетей, к которым подключен экран. Аналогично межсетевым экранам прикладного уровня, доставка трафика из одной сети в другую определяется набором правил политики. Если правило не разрешает явным образом определенный трафик, то соответствующие пакеты будут отклонены или аннулированы межсетевым экраном.

Правила политики усиливаются посредством использования фильтров пакетов. Фильтры изучают пакеты и определяют, является ли трафик разрешенным, согласно правилам политики и состоянию протокола (проверка с учетом состояния). Если протокол приложения функционирует через TCP, определить состояние относительно просто, так как TCP сам по себе поддерживает состояния. Это означает, что когда протокол находится в определенном состоянии, разрешена передача только определенных пакетов. Рассмотрим в качестве примера последовательность установки соединения. Первый ожидаемый пакет - пакет SYN. Межсетевой экран обнаруживает этот пакет и переводит соединение в состояние SYN. В данном состоянии ожидается один из двух пакетов - либо SYN ACK (опознавание пакета и разрешение соединения) или пакет RST (сброс соединения по причине отказа в соединении получателем). Если в данном соединении появятся другие пакеты, межсетевой экран аннулирует или отклонит их, так как они не подходят для данного состояния соединения, даже если соединение разрешено набором правил.

Если протоколом соединения является UDP, межсетевой экран с пакетной фильтрацией не может использовать присущее протоколу состояние, вместо чего отслеживает состояние трафика UDP. Как правило, межсетевой экран принимает внешний пакет UDP и ожидает входящий пакет от получателя, соответствующий исходному пакету по адресу и порту, в

течение определенного времени. Если пакет принимается в течение этого отрезка времени, его передача разрешается. В противном случае межсетевой экран определяет, что трафик UDP не является ответом на запрос, и аннулирует его.

При использовании межсетевого экрана с пакетной фильтрацией соединения не прерываются на межсетевом экране (см. рис. 2), а направляются непосредственно к конечной системе. При поступлении пакетов межсетевой экран выясняет, разрешен ли данный пакет и состояние соединения правилами политики. Если это так, пакет передается по своему маршруту. В противном случае пакет отклоняется или аннулируется.

Рис. 2. Передача трафика через межсетевой экран с фильтрацией пакетов

Гибридные межсетевые экраны

Как и многие другие устройства, межсетевые экраны изменяются и совершенствуются с течением времени, т. е. эволюционируют. Производители межсетевых экранов прикладного уровня в определенный момент пришли к выводу, что необходимо разработать метод поддержки протоколов, для которых не существует определенных модулей доступа. Вследствие этого увидела свет технология модуля доступа Generic Services Proxy (GSP). GSP разработана для поддержки модулями доступа прикладного уровня других протоколов, необходимых системе безопасности и при работе сетевых администраторов. В действительности GSP обеспечивает работу межсетевых экранов прикладного уровня в качестве экранов с пакетной фильтрацией.

Производители межсетевых экранов с пакетной фильтрацией также добавили некоторые модули доступа в свои продукты для обеспечения более высокого уровня безопасности некоторых широко распространенных протоколов. На сегодняшний день многие межсетевые экраны с пакетной фильтрацией поставляются с модулем доступа SMTP.

В то время как базовая функциональность межсетевых экранов обоих типов осталась прежней, (что является причиной большинства "слабых мест" этих устройств), сегодня на рынке присутствуют гибридные межсетевые экраны. Практически невозможно найти межсетевой экран, функционирование которого построено исключительно на прикладном уровне или фильтрации пакетов. Это обстоятельство отнюдь не является недостатком, так как оно позволяет администраторам, отвечающим за безопасность, настраивать устройство для работы в конкретных условиях.