

Министерство образования и науки Российской Федерации
ГОУ ВПО «Тамбовский государственный технический университет»

**С.А. ВАСИЛЬЕВ, В.Е. ПОДОЛЬСКИЙ,
И.В. МИЛОВАНОВ, В.И. ЛОСКУТОВ**

**ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ В САПР
ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ
И КОМПЬЮТЕРНАЯ ГРАФИКА**

*Рекомендовано Учёным советом университета
в качестве учебного пособия
для студентов 3–4 курсов специальности 230104 дневного отделения*



Тамбов
Издательство ТГТУ
2008

УДК 519.68
ББК 32.073
И741

Рецензенты:

Доктор технических наук, профессор
Ю.В. Литовка

Доктор технических наук, профессор
А.Г. Ткачёв

И741 Информационные технологии в САПР. Вычислительные сети и компьютерная графика : учебное пособие / С.А. Васильев, В.Е. Подольский, И.В. Милованов, В.И. Лоскутов. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2008. – 80 с. – 100 экз. – ISBN 978-5-8265-0791-9.

Изложен материал по вычислительным сетям и основным алгоритмам растровой $2D$ и $3D$ графики, который может быть использован для организации высокоэффективной САПР, а также их научно-исследовательской работе.

Предназначено для студентов 3–4 курсов дневной формы обучения по специальности 230104.

УДК 519.68
ББК 32.073

ISBN 978-5-8265-0791-9

© ГОУ ВПО «Тамбовский государственный
технический университет» (ТГТУ), 2008

Учебное издание

ВАСИЛЬЕВ Сергей Александрович,
ПОДОЛЬСКИЙ Владимир Ефимович,
МИЛОВАНОВ Игорь Викторович,
ЛОСКУТОВ Вячеслав Иванович

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В САПР
ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ
И КОМПЬЮТЕРНАЯ ГРАФИКА**

Учебное пособие

Редактор З.Г. Чернова
Инженер по компьютерному макетированию Т.Ю. Зотова

Подписано в печать 30.12.2008
Формат 60 × 84 / 16. 4,65 усл. печ. л. Тираж 100 экз. Заказ № 605

Издательско-полиграфический центр ТГТУ
392000, Тамбов, Советская, 106, к. 14

ПРЕДИСЛОВИЕ

Учебное пособие предназначено для студентов специальностей «Системы автоматизированного проектирования» и других, изучающих курсы «Сети ЭВМ и телекоммуникации» и «Сетевые технологии».

Самая главная функция компьютера – обработка информации. Особо можно выделить обработку информации, связанную с изображениями. Она разделяется на три основных направления: компьютерная графика, обработка изображений и распознавание образов.

Современные устройства вывода графической информации в основном имеют растровую организацию, т.е. в качестве экрана имеется матрица ячеек (пикселей). Совокупность пикселей различного цвета формирует изображение. Для организации растровых изображений существует целая наука – компьютерная растровая графика.

Данное пособие посвящено алгоритмическим основам компьютерной растровой графики.

Компьютерная графика в настоящее время уже вполне сформировалась как наука и используется почти во всех научных и инженерных дисциплинах для наглядности восприятия и передачи информации. Знание её основ в наше время необходимо любому учёному или инженеру. Компьютерная графика властно вторгается в бизнес, медицину, рекламу и индустрию развлечений.

Материал данного пособия базируется на реальном курсе компьютерной графики, что читается в Тамбовском государственном техническом университете на кафедре САПР на протяжении последних нескольких лет.

Пособие в целом может быть использовано для изучения основных тем вводного курса компьютерной графики.

Часть I

ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

1. ВВЕДЕНИЕ В ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

1.1. ПОНЯТИЕ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ. КЛАССИФИКАЦИЯ

Коммуникационная сеть – это система, состоящая из объектов, осуществляющих функции генерации, преобразования, хранения и потребления продукта, называемых пунктами (узлами) сети, и линий передачи (связей, коммуникаций, соединений), осуществляющих передачу продукта между пунктами.

Отличительная особенность коммуникационной сети – большие расстояния между пунктами по сравнению с геометрическими размерами участков пространства, занимаемых пунктами. В качестве продукта могут фигурировать информация, энергия, масса, и соответственно различают группы сетей информационных, энергетических, вещественных. В группах сетей возможно разделение на подгруппы. Так, среди вещественных сетей могут быть выделены сети транспортные, водопроводные, производственные и др. В данной части пособия рассматриваются информационные сети.

Информационная сеть – коммуникационная сеть, в которой продуктом генерирования, переработки, хранения и использования является информация.

Вычислительная сеть – информационная сеть, в состав которой входит вычислительное оборудование. Компонентами вычислительной сети могут быть ЭВМ и периферийные устройства, являющиеся источниками и приёмниками данных, передаваемых по сети. Эти компоненты составляют оконечное оборудование данных (ООД или DTE – Data Terminal Equipment). В качестве ООД могут выступать ЭВМ, принтеры, плоттеры и другое вычислительное, измерительное и исполнительное оборудование автоматических и автоматизированных систем. Собственно пересылка данных происходит с помощью сред и средств, объединяемых под названием среда передачи данных.

Подготовка данных, передаваемых или получаемых ООД от среды передачи данных, осуществляется функциональным блоком, называемым *аппаратурой окончания канала данных* (АКД или DCE – Data Circuit-Terminating Equipment). АКД может быть конструктивно отдельным или встроенным в ООД блоком. ООД и АКД вместе представляют собой *станцию данных*, которую часто называют узлом сети. Примером АКД может служить модем.

Вычислительные сети классифицируются по ряду признаков.

В зависимости от расстояний между связываемыми узлами различают вычислительные сети:

– *территориальные** (WAN – Wide Area Network), охватывающие значительное географическое пространство; среди территориальных сетей можно выделить сети региональные и глобальные, имеющие соответственно региональные или глобальные масштабы;

– *локальные* (ЛВС** LAN – Local Area Network), охватывающие ограниченную территорию (обычно в пределах удалённости станций не более чем на несколько десятков или сотен метров друг от друга, реже на 1...2 км);

– *корпоративные (масштаб предприятия)* – совокупность связанных между собой ЛВС, охватывающих территорию, на которой размещено одно предприятие. Локальные и корпоративные вычислительные сети – основной вид вычислительных сетей, используемых в системах автоматизированного проектирования (САПР);

– особо выделяют глобальную сеть Internet (реализованная в ней информационная служба World Wide Web (WWW) переводится на русский язык как всемирная паутина); это сеть сетей со своей технологией. В Internet существует понятие *интрасетей* (Intranet) – корпоративных сетей в рамках Internet.

Различают интегрированные сети, неинтегрированные сети и подсети. *Интегрированная вычислительная сеть (интерсеть)* представляет собой взаимосвязанную совокупность многих вычислительных сетей, которые в интрасети называются подсетями.

В автоматизированных системах крупных предприятий подсети включают вычислительные средства отдельных подразделений. Интерсети нужны для объединения таких подсетей, а также для объединения технических средств автоматизированных систем проектирования и производства в единую систему комплексной автоматизации (СІМ – Computer Integrated Manufacturing). Обычно интерсети приспособлены для различных видов связи: телефонии, электронной почты, передачи видеoinформации, цифровых данных и т.п., и в этом случае они называются *сетями интегрального обслуживания*.

Подсети в интерсетях объединяются в соответствии с выбранной топологией с помощью *блоков взаимодействия*.

В зависимости от топологии соединений узлов различают сети шинной (магистральной), кольцевой, звёздной, иерархической, произвольной структуры.

Среди ЛВС наиболее распространены (рис. 1.1):

– *шинная (bus)* – локальная сеть, в которой связь между любыми двумя станциями устанавливается через один общий путь и данные, передаваемые любой станцией, одновременно становятся доступными для всех других станций, подключенных к этой же среде передачи данных (последнее свойство называют *широковещательностью*);

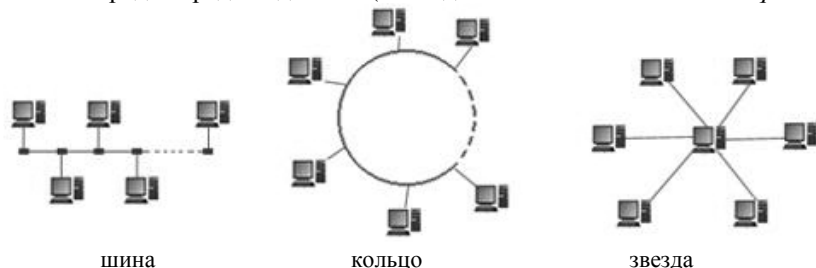


Рис. 1.1. Основные топологические структуры локальных вычислительных сетей

– *кольцевая (ring)* – узлы связаны кольцевой линией передачи данных (к каждому узлу подходят только две линии); данные, проходя по кольцу, поочерёдно становятся доступными всем узлам сети;

– *звёздная (star)* – имеется центральный узел, от которого расходятся линии передачи данных к каждому из остальных узлов.

В зависимости от способа управления различают сети:

– «*клиент/сервер*» – в них выделяется один или несколько узлов (их название – серверы), выполняющих в сети управляющие или специальные обслуживающие функции, а остальные узлы (клиенты) являются терминальными, в них работают пользователи;

– *одноранговые* – в них все узлы равноправны; в одноранговых сетях могут выполнять функции и клиента, и сервера;

– *сетевые* – пользователь имеет лишь дешёвое оборудование для обращения к удалённым компьютерам, а сеть обслуживает заказы на выполнение вычислений и получения информации. Подобные компьютеры называют тонкими клиентами или сетевыми компьютерами.

В зависимости от того, одинаковые или неодинаковые ЭВМ применяют в сети, различают сети однотипных ЭВМ, называемые *однородными*, и разнотипных ЭВМ – *неоднородными (гетерогенными)*. В крупных автоматизированных системах, как правило, сети оказываются неоднородными.

В зависимости от прав собственности на сети последние могут быть *сетями общего пользования (public)* или *частными (private)*. Среди сетей общего пользования выделяют телефонные сети ТФОП (PSTN – Public Switched Telephone Network) и сети передачи данных (PSDN – Public Switched Data Network).

Сети также различают в зависимости от используемых в них протоколов и по способам коммутации.

1.2. СПОСОБЫ КОММУТАЦИИ

Под *коммутацией данных* понимается их передача, при которой канал передачи данных может использоваться попеременно для обмена информацией между различными пунктами информационной сети в отличие от связи через некоммутируемые каналы, обычно закреплённые за определёнными абонентами.

Различают следующие способы коммутации данных:

– *коммутация каналов* – осуществляется соединение двух или более станций данных и обеспечивается монопольное использование канала передачи данных до тех пор, пока соединение не будет разорвано;

– *коммутация сообщений* – характеризуется тем, что создание физического канала между оконечными узлами необязательно и пересылка сообщений происходит без нарушения их целостности; вместо физического канала имеется виртуальный канал, состоящий из физических участков, и между участками возможна буферизация сообщения;

– *коммутация пакетов* – сообщение передаётся по виртуальному каналу, но оно разделяется на пакеты, при этом канал передачи данных занят только во время передачи пакета (без нарушения его целостности) и по её завершении освобождается для передачи других пакетов.

Коммутация каналов может быть пространственной и временной.

Пространственный коммутатор размера $N \times M$ представляет собой сетку (матрицу), в которой N входов подключены к горизонтальным шинам, а M выходов – к вертикальным (рис. 1.2).

В узлах сетки имеются коммутирующие элементы, причем в каждом столбце сетки может быть открыто не более чем по одному элементу. Если $N < M$, то коммутатор может обеспечить соединение каждого входа с не менее чем одним выходом; в противном случае коммутатор называется блокирующим, т.е. не обеспечивающим соединения любого входа с одним из выходов. Обычно применяются коммутаторы с равным числом входов и выходов $N \times N$.

Недостаток рассмотренной схемы – большое число коммутирующих элементов в квадратной матрице, равное N^2 . Для устранения этого недостатка применяют многоступенные коммутаторы. Например, схема трёхступенного коммутатора 6×6 имеет вид, представленный на рис. 1.3.

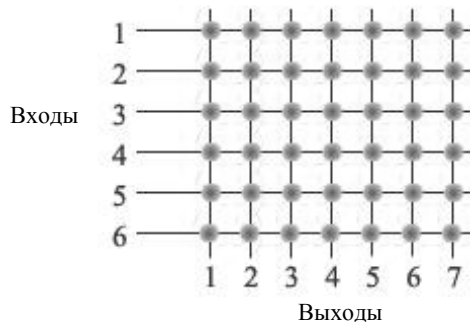


Рис. 1.2. Матрица пространственного коммутатора

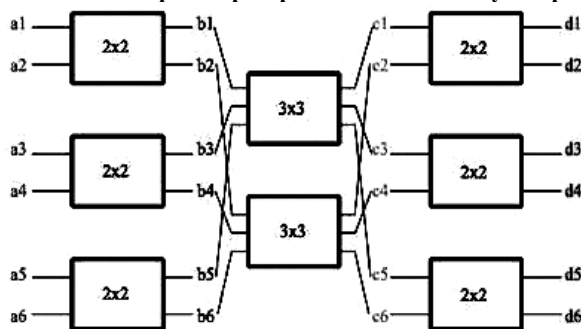


Рис. 1.3. Схема трёхступенного пространственного коммутатора

Достаточным условием отсутствия блокировок входов является равенство $k > 2 \times n - 1$. Здесь k – число блоков в промежуточном каскаде, $n = N/p$, p – число блоков во входном каскаде. В приведённой на рис. 1.3 схеме это условие не выполнено, поэтому блокировки возможны.

В многоступенных коммутаторах существенно уменьшено число переключательных элементов за счёт некоторого увеличения задержки. Например, при замене одноступенного коммутатора 1000×1000 трёхступенным с $n = 22$ и $k = 43$ число переключателей уменьшается с 10^6 до $2 \cdot 46 \cdot 22 \cdot 43 + 43 \cdot 46 \cdot 46$, т.е. примерно до $0,186 \cdot 10^6$.

Временной коммутатор построен на основе буферной памяти, запись производится в её ячейки последовательным опросом входов, а коммутация осуществляется благодаря считыванию данных на выходы из нужных ячеек памяти. При этом происходит задержка на время одного цикла «запись–чтение». В настоящее время преимущественно используются временная или смешанная коммутация.

Во многих случаях наиболее эффективной оказывается коммутация пакетов. Во-первых, ускоряется передача данных в сетях сложной конфигурации за счёт того, что возможна параллельная передача пакетов одного сообщения на разных участках сети; во-вторых, при появлении ошибки требуется повторная передача короткого пакета, а не всего длинного сообщения. Кроме того, ограничение сверху на размер пакета позволяет обойтись меньшим объёмом буферной памяти в промежуточных узлах на маршрутах передачи данных в сети.

В сетях коммутации пакетов различают два режима работы: режим виртуальных каналов (другое название – связь с установлением соединения) и дейтаграммный режим (связь без установления соединения).

В *режиме виртуальных каналов* пакеты одного сообщения передаются в естественном порядке по устанавливаемому маршруту. При этом в отличие от коммутации каналов линии связи могут разделяться многими сообщениями, когда попеременно по каналу передаются пакеты разных сообщений (это так называемый режим временного мультиплексирования, иначе TDM – Time Division Method), или задерживаться в промежуточных буферах. Предусматривается контроль правильности передачи данных путём посылки от получателя к отправителю подтверждающего сообщения – положительной квитанции. Этот контроль возможен как во всех промежуточных узлах маршрута, так и только в конечном узле. Он может осуществляться *старт-стопным* способом, при котором отправитель до тех пор не передаёт следующий пакет, пока не получит подтверждения о правильной передаче предыдущего пакета, или способом *передачи «в окне»*. Окно может включать N пакетов, и возможны задержки в получении подтверждений на протяжении окна. Так, если произошла ошибка при передаче, т.е. отправитель получает отрицательную квитанцию относительно пакета с номером K , то нужна повторная передача, и она начинается с пакета K .

Например, в сетях можно использовать переменный размер окна. Так, в соответствии с рекомендацией документа RFC-793 время ожидания подтверждений вычисляется по формуле

$$T_{\text{ож}} = 2T_{\text{ср}}, \quad (1)$$

где $T_{\text{ср}} = 0,9T_i$, T_i – усреднённое значение времени прохода пакета до получателя и обратно; T_i – результат очередного измерения этого времени.

В *дейтаграммном режиме* сообщение делится на дейтаграммы. Дейтаграмма – часть информации, передаваемая независимо от других частей одного и того же сообщения в вычислительных сетях с коммутацией пакетов. Дейтаграммы одного и того же сообщения могут передаваться в сети по разным маршрутам и поступать к адресату в произвольной последователь-

ности, что может послужить причиной блокировок сети. На внутренних участках маршрута контроль правильности передачи не предусмотрен, и надёжность связи обеспечивается лишь контролем на конечном узле.

Блокировкой сети в дейтаграммном режиме называется такая ситуация, когда в буферную память узла вычислительной сети поступило столько пакетов разных сообщений, что эта память оказывается полностью занятой. Следовательно, она не может принимать другие пакеты и не может освободиться от уже принятых, так как это возможно только после поступления всех дейтаграмм сообщения.

1.3. ПРОТОКОЛЫ

Протокол – это набор семантических и синтаксических правил, определяющий поведение функциональных блоков сети при передаче данных. Другими словами, протокол – это совокупность соглашений относительно способа представления данных, обеспечивающего их передачу в нужных направлениях и правильную интерпретацию данных всеми участниками процесса информационного обмена.

Поскольку информационный обмен – процесс многофункциональный, то протоколы делятся на уровни. К каждому уровню относится группа родственных функций. Для правильного взаимодействия узлов различных вычислительных сетей их архитектура должна быть открытой. Этим целям служат унификация и стандартизация в области телекоммуникаций и вычислительных сетей.

Унификация и стандартизация протоколов выполняются рядом международных организаций, что наряду с разнообразием типов сетей породило большое число различных протоколов. Наиболее широко распространёнными являются протоколы, разработанные для сети ARPANET и применяемые в глобальной сети Internet, протоколы открытых систем Международной организации по стандартизации (ISO–International Standard Organization), протоколы Международного телекоммуникационного союза (International Telecommunication Union–ITU, ранее называвшегося ССИТТ) и протоколы Института инженеров по электротехнике и электронике (IEEE – Institute of Electrical and Electronics Engineers). Протоколы сети Internet объединяют под названием TCP/IP. Протоколы ISO являются семиуровневыми и известны как протоколы *базовой эталонной модели взаимосвязи открытых систем* – ЭМВОС).

1.4. ЭТАЛОННАЯ МОДЕЛЬ ВЗАИМОСВЯЗИ ОТКРЫТЫХ СИСТЕМ

Базовая ЭМВОС – это модель, принятая ISO для описания общих принципов взаимодействия информационных систем. ЭМВОС признана всеми международными организациями как основа для стандартизации протоколов информационных сетей.

В ЭМВОС информационная сеть рассматривается как совокупность функций, которые делятся на группы, называемые *уровнями*. Разделение на уровни позволяет вносить изменения в средства реализации одного уровня без перестройки средств других уровней, что значительно упрощает и удешевляет модернизацию средств по мере развития техники.

ЭМВОС содержит семь уровней, их номера, названия и выполняемые функции которых приведены ниже.

7-й уровень – *прикладной (Application)*: включает средства управления прикладными процессами; эти процессы могут объединяться для выполнения поставленных заданий, обмениваться между собой данными. Другими словами, на этом уровне определяются и оформляются в блоки те данные, которые подлежат передаче по сети. Уровень включает, например, такие средства для взаимодействия прикладных программ, как приём и хранение пакетов в «почтовых ящиках» (mail-box).

6-й уровень – *представительный (Presentation)*: реализуются функции представления данных (кодирование, форматирование, структурирование). Например, на этом уровне выделенные для передачи данные преобразуются из кода EBCDIC в ASCII и т.п.

5-й уровень – *сеансовый (Session)*: предназначен для организации и синхронизации диалога, ведущегося объектами (станциями) сети. На этом уровне определяются тип связи (дуплекс или полудуплекс), начало и окончание заданий, последовательность и режим обмена запросами и ответами взаимодействующих партнеров.

4-й уровень – *транспортный (Transport)*: предназначен для управления сквозными каналами в сети передачи данных; на этом уровне обеспечивается связь между конечными пунктами (в отличие от следующего сетевого уровня, на котором обеспечивается передача данных через промежуточные компоненты сети). К функциям транспортного уровня относятся мультиплексирование и демultipлексирование (сборка-разборка пакетов), обнаружение и устранение ошибок в передаче данных, реализация заказанного уровня услуг (например, заказанной скорости и надёжности передачи).

3-й уровень – *сетевой (Network)*: на этом уровне происходит формирование пакетов по правилам тех промежуточных сетей, через которые проходит исходный пакет, и *маршрутизация* пакетов, т.е. определение и реализация маршрутов, по которым передаются пакеты. Другими словами, маршрутизация сводится к образованию логических каналов. *Логическим каналом* называется виртуальное соединение двух или более объектов сетевого уровня, при котором возможен обмен данными между этими объектами. Ещё одной важной функцией сетевого уровня после маршрутизации является контроль нагрузки на сеть с целью предотвращения перегрузок, отрицательно влияющих на работу сети.

2-й уровень – *канальный (Link, уровень звена данных)*: предоставляет услуги по обмену данными между логическими объектами предыдущего сетевого уровня и выполняет функции, связанные с формированием и передачей кадров, обнаружением и исправлением ошибок, возникающих на следующем, физическом уровне. *Кадром* называется пакет канального уровня, поскольку пакет на предыдущих уровнях может состоять из одного или многих кадров.

1-й уровень – *физический (Physical)*: предоставляет механические, электрические, функциональные и процедурные средства для установления, поддержания и разъединения логических соединений между логическими объектами канального уровня; реализует функции передачи битов данных через физические среды. Именно на физическом уровне осуществляются представление информации в виде электрических или оптических сигналов, преобразования формы сигналов, выбор параметров физических сред передачи данных.

В конкретных случаях может возникать потребность в реализации лишь части названных функций, тогда соответственно в сети имеется лишь часть уровней. Так, в простых (неразветвлённых) ЛВС отпадает необходимость в средствах сетевого и транспортного уровней. В то же время сложность функций канального уровня делает целесообразным его разделение в ЛВС на два по-

уровня: *управление доступом к каналу* (MAC – Medium Access Control) и *управление логическим каналом* (LLC – Logical Link Control). К подуровню LLC в отличие от подуровня MAC относится часть функций канального уровня, не связанных с особенностями передающей среды.

Передача данных через разветвленные сети происходит при использовании *инкапсуляции/декапсуляции* порций данных. Так, сообщение, пришедшее на транспортный уровень, делится на сегменты, которые получают заголовки и передаются на сетевой уровень. Сегментом обычно называют пакет транспортного уровня. Сетевой уровень организует передачу данных через промежуточные сети. Для этого сегмент может быть разделен на части (пакеты), если сеть не поддерживает передачу сегментов целиком. Пакет снабжается своим сетевым заголовком (т.е. происходит инкапсуляция). При передаче между узлами промежуточной ЛВС требуется инкапсуляция пакетов в кадры с возможной разбивкой пакета. Приёмник декапсулирует сегменты и восстанавливает исходное сообщение.

2. КАНАЛЫ ПЕРЕДАЧИ ДАННЫХ

2.1. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Среда передачи данных – совокупность линий передачи данных и блоков взаимодействия (т.е. сетевого оборудования, не входящего в станции данных), предназначенных для передачи данных между станциями данных. Среда передачи данных могут быть общего пользования или выделенными для конкретного пользователя.

Линия передачи данных – средства, которые используются в информационных сетях для распространения сигналов в нужном направлении. Характеристиками линий передачи данных являются зависимости затухания сигнала от частоты и расстояния. Затухание принято оценивать в децибелах:

$$1 \text{ дБ} = 10 \lg(P_1/P_2) \text{ дБ}, \quad (2.1)$$

где P_1 и P_2 – мощности сигнала на входе и выходе линии, соответственно.

При заданной длине можно говорить о полосе пропускания (полосе частот) линии. Полоса пропускания связана со скоростью передачи информации. Различают бодовую (модуляционную) и информационную скорости. Бодовая скорость измеряется в бодах, т.е. числом изменений дискретного сигнала в единицу времени, а информационная – числом битов информации, переданных в единицу времени. Именно бодовая скорость определяется полосой пропускания линии.

Если на бодовом интервале (между соседними изменениями сигнала) передаётся N бит, то число градаций модулируемого параметра несущей равно 2^N . Например, при числе градаций 16 и скорости 1200 бод одному боду соответствует 4 бит/с и информационная скорость составит 4800 бит/с.

Максимально возможная информационная скорость V связана с полосой пропускания F канала связи формулой Хартли-Шеннона (предполагается, что одно изменение величины сигнала приходится на $\log_2 k$ бит, где k – число возможных дискретных значений сигнала)

$$V = 2F \log_2 k \text{ бит/с}, \quad (2.2)$$

так как

$$V = \log_2 k/t, \quad (2.3)$$

где t – длительность переходных процессов, приблизительно равная $3 \cdot \text{ТВ}$; $\text{ТВ} = 1/(2pF)$; $k = 1 + A$, A – отношение сигнал/помеха.

Канал (канал связи) – средства односторонней передачи данных. Примером канала может быть полоса частот, выделенная одному передатчику при радиосвязи. В некоторой линии можно образовать несколько каналов связи, по каждому из которых передаётся своя информация. При этом говорят, что линия разделяется между несколькими каналами. Существуют два метода разделения линии передачи данных: временное мультиплексирование (иначе разделение по времени или TDM), при котором каждому каналу выделяется некоторый квант времени, и частотное разделение (FDM – Frequency Division Method), при котором каналу выделяется некоторая полоса частот.

Канал передачи данных – средства двустороннего обмена данными, включающие АКД и линию передачи данных.

По природе физической среды передачи данных различают каналы передачи данных на оптических линиях связи, проводных (медных) линиях связи и беспроводные. В свою очередь, медные каналы могут быть представлены коаксиальными кабелями и витыми парами, а беспроводные – радио- и инфракрасными каналами.

В зависимости от способа представления информации электрическими сигналами различают аналоговые и цифровые каналы передачи данных. В аналоговых каналах для согласования параметров среды и сигналов применяют амплитудную, частотную, фазовую и квадратурно-амплитудную модуляции. В цифровых каналах для передачи данных используют самосинхронизирующиеся коды, а для передачи аналоговых сигналов – кодово-импульсную модуляцию.

В зависимости от направления передачи различают каналы симплексные (односторонняя передача), дуплексные (возможность одновременной передачи в обоих направлениях) и полудуплексные (возможность попеременной передачи в двух направлениях).

В зависимости от числа каналов связи в аппаратуре передачи данных различают одно- и многоканальные средства передачи. В локальных вычислительных сетях и в цифровых каналах передачи данных обычно используют временное мультиплексирование, в аналоговых каналах – частотное разделение.

К передаче информации имеют прямое отношение телефонные сети, вычислительные сети передачи данных, спутниковые системы связи, системы содовой радиосвязи.

2.2. ПРОВОДНЫЕ ЛИНИИ СВЯЗИ

В вычислительных сетях проводные линии связи представлены коаксиальными кабелями и витыми парами проводов.

Используются коаксиальные кабели: «толстый» диаметром 12,5 мм и «тонкий» диаметром 6,25 мм. «Толстый» кабель имеет меньшее затухание, лучшую помехозащищённость, что обеспечивает возможность работы на больших расстояниях, но он плохо гнётся, что затрудняет прокладку соединений в помещениях, и дороже «тонкого».

Существуют экранированные (STP – Shielded Twist Pair) и неэкранированные (UTP – Unshielded Twist Pair) витые пары проводов. Неэкранированные витые пары имеют несколько категорий (типов). Обычный телефонный кабель – пара категории 1. Пара категории 2 может использоваться в сетях с пропускной способностью до 4 Мбит/с. Для сетей Ethernet (точнее, для её варианта с названием 10Base-T) разработана пара категории 3, а для сетей Token Ring – пара категории 4. Наиболее совершенной является витая пара категории 6, которая применима при частотах до 1 ГГц. В паре категории 6 проводник представлен медными жилами диаметром 0,51 мм, навитыми по определённой технологии и заключёнными в термостойкую изолирующую оболочку. В высокоскоростных ЛВС на UTP длины соединений обычно не превышают 100 м. Затухание на 1 ГГц и при длине 100 м составляет около 24 дБ, при 10 МГц и 100 м – около 7 дБ.

Витые пары иногда называют сбалансированной линией в том смысле, что в двух проводах линии передаются одни и те же уровни сигнала (по отношению к земле), но разной полярности. При приёме воспринимается разность сигналов, называемая парафазным сигналом. Синфазные помехи при этом самокомпенсируются.

2.3. МОДУЛЯЦИЯ

Амплитудная модуляция. При амплитудной модуляции на входы модулятора поступают сигнал V и несущая U . Например, если сигнал есть гармоническое колебание

$$V = V_m \sin(Wt + j) \quad (2.4)$$

с амплитудой V_m , частотой W и фазой j , то на выходе нелинейного элемента в модуляторе будут модулированные колебания

$$U_{AM} = U_m (1 + m \sin(Wt + j)) \cdot \sin(vt + y), \quad (2.5)$$

где $m = V_m/U_m$ – коэффициент модуляции. На выходе модулятора в спектре сигнала присутствуют несущая частота v и две боковые частоты $v + W$ и $v - W$. Если сигнал занимает некоторую полосу частот, то в спектре модулированного колебания появятся две боковые полосы, как это показано на рис. 2.1.

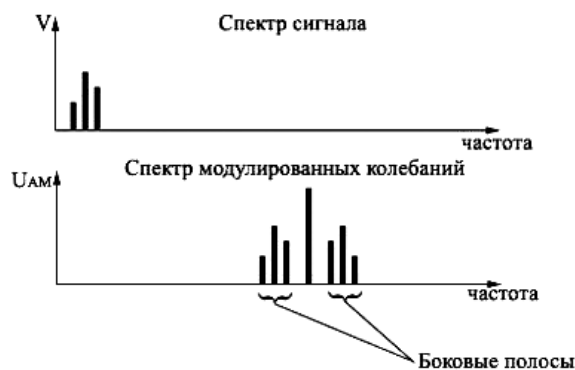


Рис. 2.1. Спектры модулирующего и модулированного сигналов при АМ

При амплитудной модуляции во избежание искажений, называемых качанием фронта, нужно выполнение условия $v \gg W$, где v и W – соответственно несущая и модулирующая частоты. Соблюдение этого условия при стандартной (для средне-скоростной аппаратуры передачи данных) несущей частоте 1700 Гц не может обеспечить информационные скорости выше 300 бит/с. Поэтому в модемах применяют дополнительное преобразование частоты: сначала производят модуляцию несущей, имеющей повышенную частоту, например $F_{нд} = 10$ кГц, затем с помощью фильтра выделяют спектр модулированного сигнала и с помощью преобразователя частоты переносят модулирующие колебания на промежуточную частоту, например 1700 Гц. Тогда при боковых полосах до 1400 Гц спектр сигнала согласуется с полосой пропускания телефонных линий. Однако достигаемые при этом скорости передачи данных остаются невысокими.

Скорости передачи повышаются с помощью квадратурно-амплитудной или фазовой модуляции за счёт того, что вместо двоичных модулирующих сигналов используются дискретные сигналы с большим числом возможных значений.

Частотная и фазовая модуляции. В сравнительно простых модемах применяют частотную модуляцию (FSK – Frequency Shift Keying) со скоростями передачи до 1200 бит/с. Так, если необходима дуплексная связь по двухпроводной линии, то возможно представление 1 и 0 в вызывном модеме частотами 980 и 1180 Гц, соответственно, а в ответном модеме – 1650 и 1850 Гц. При этом скорость передачи составляет 300 бод.

Обычно для передачи сигнала об ошибке от приёмника к передатчику нужен канал обратной связи. При этом требования к скорости передачи данных по обратному каналу могут быть невысокими. Тогда в полосе частот телефонного канала образуют обратный канал с ЧМ, по которому со скоростью 75 бит/с передают 1 частотой 390 Гц и 0 частотой 450 Гц.

Фазовая модуляция (PSK – Phase Shift Keying) двумя уровнями сигнала (1 и 0) осуществляется переключением между двумя несущими, сдвинутыми на полпериода друг относительно друга. Другой вариант PSK изменение фазы на $\pi/2$ в каждом такте при передаче нуля и на $3/4 \cdot \pi$, если передается единица.

Квадратурно-амплитудная модуляция. Квадратурно-амплитудная модуляция (QAM – Quadrature Amplitude Modulation, её также называют квадратурно-импульсной) основана на передаче одним элементом модулированного сигнала n бит информации, где $n = 4..8$ (т.е. используются 16... 256 дискретных значений амплитуды). Однако для надёжного различения этих значений амплитуды требуется малый уровень помех (отношение сигнал/помеха не менее 12 дБ при $n = 4$).

При меньших отношениях сигнал/помеха лучше применять фазовую модуляцию с четырьмя или восемью дискретными значениями фазы для представления соответственно 2 или 3 бит информации. Тогда при скорости модуляции в 1200 бод (т.е. 1200 элементов аналогового сигнала в секунду, где элемент – часть сигнала между возможными сменами фаз) и четырёхфаз-

ной модуляции скорость передачи данных равна 2400 бит/с. Используются также скорости передачи 4800 бит/с (при скорости модуляции 1600 бод и восьмифазной модуляции), 9600 бит/с и более при комбинации фазовой и амплитудной модуляций.

Кодово-импульсная модуляция. Кодово-импульсная модуляция (КИМ или PCM – Pulse Code Modulation) используется для передачи аналоговых сигналов по цифровым каналам связи.

Этот вид модуляции сводится к измерению амплитуды аналогового сигнала в моменты времени, отстоящие друг от друга на dt , и к кодированию этих амплитуд цифровым кодом. Величина dt определяется по теореме Котельникова: для неискажённой передачи нужно иметь не менее двух отсчётов на период колебаний, соответствующий высшей составляющей в частотном спектре сигнала.

При преобразовании амплитуды A аналогового сигнала в цифровой код K желательно учитывать нелинейность амплитудных характеристик приборов и иметь зависимость K от A монотонно убывающей с ростом амплитуды.

Разновидностями КИМ являются дельта-модуляция (ДМ), дифференциальная ДМ (ДДМ) и адаптивная ДМ (АДДМ). В них передаются разности амплитуд A_1 и A_2 соседних отсчётов. При этом в ДМ A_1 – амплитуда на входе модулятора, а A_2 – амплитуда отсчёта, которая соответствует переданному сигналу в предыдущем временном такте. Для представления разности используется всего 1 бит (т.е. передаётся знак разности), поэтому нужна достаточно высокая частота отсчётов, чтобы не было «запаздывания» изменений передаваемого сигнала по сравнению с реальными изменениями.

ДДМ отличается от ДМ тем, что знак разности $A_1 - A_2$ передаётся только в момент пересечения величиной A_1 одного из уровней квантования. В АДДМ шаги отсчётов адаптируются к динамике изменения величины сигнала.

2.4. БЕСПРОВОДНЫЕ КАНАЛЫ СВЯЗИ

В беспроводных каналах передача информации осуществляется на основе распространения радиоволн. Сведения о диапазонах электромагнитных колебаний, используемых в беспроводных и оптических каналах связи, приведены в табл. 2.1.

Чем выше рабочая частота, тем больше ёмкость (число каналов) системы связи, но тем меньше предельные расстояния, на которых возможна прямая передача между двумя пунктами без ретрансляторов. Первая из причин и порождает тенденцию к освоению новых более высокочастотных диапазонов.

Радиоканалы входят необходимой составной частью в спутниковые и радиорелейные системы связи, применяемые в территориальных сетях, в сотовые системы мобильной связи. Они используются в качестве альтернативы кабельным системам в локальных сетях и при объединении сетей отдельных офисов и предприятий в корпоративные сети. Во многих случаях применение радиоканалов оказывается более дешёвым решением по сравнению с другими вариантами.

2.1. Диапазоны электромагнитных колебаний

Диапазон	Длины волн, м	Частоты, ГГц	Применение
Дециметровый	1, ... , 0,1	0,3, ... , 3	Сотовые радиотелефоны, ТВ, спутниковая связь
Сантиметровый	0,1, ... , 0,01	3...30	Радиорелейные линии, радиоканалы в ЛВС, спутниковая связь
Миллиметровый	0,01, ... , 0,001	30...300	Радиоканалы в ЛВС
Инфракрасный	$(0,001, \dots, 7,5) \cdot 10^{-7}$	$300 \dots 4 \cdot 10^5$	Волоконно-оптические линии, каналы с мультиплексированием по длине волны
Видимый свет	$(7,5 \dots 4) \cdot 10^{-7}$	$(4 \dots 7,5) \cdot 10^5$	

Радиосвязь используется в корпоративных и локальных сетях, если затруднена прокладка других каналов связи. Радиоканал либо выполняет роль моста между подсетями (двухточечное соединение), либо является общей средой передачи данных в ЛВС по излагаемому далее методу МДКН/ОК, либо служит соединением между центральным и терминальными узлами в сети с централизованным управлением.

В первом случае (связь двух сетей) имеем двухточечное соединение с направленными антеннами, дальность в пределах прямой видимости. Мост имеет два адаптера: один для формирования сигналов для радиоканала, другой – для кабельной подсети.

В случае использования радиоканала в качестве общей среды передачи данных в ЛВС сеть называют RadioEthernet (стандарт IEEE 802.11), она обычно используется внутри зданий. В состав аппаратуры входят приёмопередатчики и антенны. Связь осуществляется на частотах от одного до нескольких гигагерц. Расстояния между узлами – несколько десятков метров.

В соответствии со стандартом IEEE 802.11 возможны два способа передачи двоичной информации в ЛВС, оба они имеют целью обеспечить защиту информации от нежелательного доступа.

Первый способ называется методом прямой последовательности (DSSS – Direct Sequence Spread Spectrum). В нём вводится избыточность – каждый бит данных представляется последовательностью из 11 элементов («чипов»). Эта последовательность создаётся по алгоритму, известному участникам связи, и потому может быть дешифрована при приёме. Избыточность повышает помехоустойчивость, что позволяет снизить требования к мощности передатчика, а для сохранения высокой скорости нужно расширять полосу пропускания.

Второй способ – метод частотных скачков (FHSS – Frequency Hopping Spread Spectrum). В этом методе полоса пропускания делится на 79 поддиапазонов. Передатчик через каждые 20 мс переключается на новый поддиапазон, причём алгоритм изменения частот известен только участникам связи и может изменяться, что и затрудняет несанкционированный доступ к данным.

В варианте использования радиоканалов для связи центрального и периферийного узлов центральный пункт имеет ненаправленную антенну, а терминальные пункты при этом используют направленные антенны. Дальность связи составляет также десятки метров, а вне помещений – сотни метров. Пример многоточечной системы: ненаправленная антенна по горизонтали, угол 30 градусов по вертикали, 5,8 ГГц – к терминалам, 2,4 ГГц – к центральному узлу, до 62 терминалов, дальность – 80 м без прямой видимости.

В оборудование беспроводных каналов передачи данных входят: сетевые адаптеры и радиомодемы, поставляемые вместе с комнатными антеннами и драйверами. Различаются способами обработки сигналов, характеризуются частотой передачи, пропускной способностью, дальностью связи.

2.5. ОПТИЧЕСКИЕ ЛИНИИ СВЯЗИ

Оптические линии связи реализуются в виде волоконно-оптических линий связи (ВОЛС). Конструкция ВОЛС – кварцевый сердечник диаметром 10 мкм, покрытый отражающей оболочкой с внешним диаметром 125...200 мкм. Типичные характеристики ВОЛС: работа на волнах 0,85...1,55 мкм, затухание 0,7 дБ/км, полоса частот – до 2 ГГц. Предельные расстояния D для передачи данных по ВОЛС (без ретрансляции) зависят от длины волны излучения L : для $L = 850$ нм имеем $D = 5$ км, а для $L = 1300$ нм $D = 50$ км, но аппаратурная реализация дороже.

ВОЛС являются основой высокоскоростной передачи данных, особенно на большие расстояния.

Именно на ВОЛС достигнуты рекордные скорости передачи информации. В экспериментальной аппаратуре с использованием метода мультиплексирования с разделением каналов по длинам волн (WDM – Wavelengths Division Multiplexing) достигнута скорость 1100 Гбит/с на расстоянии 150 км. В одной из действующих систем на основе WDM передача идёт со скоростью 40 Гбит/с на расстоянии до 320 км.

В методе WDM выделяется несколько несущих частот (каналов). Так, в последней упомянутой системе имеются 16 таких каналов вблизи частоты $4 \cdot 10^5$ ГГц, отстоящих друг от друга на 103 ГГц, в каждом канале достигается скорость 2,5 Гбит/с.

3. ЛОКАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

3.1. МЕТОДЫ ДОСТУПА

Типичная среда передачи данных в ЛВС – отрезок (сегмент) коаксиального кабеля. К нему через аппаратуру окончания канала данных подключаются узлы – компьютеры и возможно общее периферийное оборудование. Поскольку среда передачи данных общая, а запросы на сетевые обмены у узлов появляются асинхронно, то возникает проблема разделения общей среды между многими узлами, другими словами, проблема обеспечения доступа к сети.

Доступом к сети называют взаимодействие станции (узла сети) со средой передачи данных для обмена информацией с другими станциями. Управление доступом к среде – это установление последовательности, в которой станции получают доступ к среде передачи данных.

Различают случайные и детерминированные методы доступа. Среди случайных методов наиболее известен метод *множественного доступа с контролем несущей и обнаружением конфликтов* (МДКН/ОК – CSMA/CD Carrier Sense Multiple Access / Collision Detection). Этот метод основан на контроле несущей в линии передачи данных и устранении конфликтов, возникающих из-за попыток одновременного начала передачи двумя или более станциями, путём повторения попыток захвата линии через случайный отрезок времени.

МДКН/ОК является широкоэвещательным (broadcasting) методом. Все станции при применении МДКН/ОК равноправны по доступу к сети. Если линия передачи данных свободна, то в ней отсутствуют электрические колебания, что легко распознаётся любой станцией, желающей начать передачу. Такая станция захватывает линию. Любая другая станция, желающая начать передачу в некоторый момент времени t , если обнаруживает электрические колебания в линии, то откладывает передачу до момента $t + t_d$, где t_d – задержка.

Различают настойчивый и ненастойчивый МДКН/ОК в зависимости от того, как определяется t_d . В первом случае попытка захвата канала происходит сразу после его освобождения, что допустимо при слабой загрузке сети. При заметной загрузке велика вероятность того, что несколько станций будут претендовать на доступ к сети сразу после её освобождения, и, следовательно, конфликты станут частыми. В ненастойчивом МДКН/ОК задержка t_d является случайной величиной.

При работе сети каждая станция анализирует адресную часть передаваемых по сети кадров с целью обнаружения и приёма кадров, предназначенных для неё.

Алгоритмы приёма и передачи данных в одном из узлов при МДКН/ОК представлены на рис. 3.1.

Конфликтом называется ситуация, при которой две или более станции «одновременно» пытаются захватить линию. Понятие «одновременность событий» в связи с конечностью скорости распространения сигналов по линии конкретизируется как отстояние событий во времени не более чем на величину $2d$, называемую *окном столкновений*, где d – время прохождения сигналов по линии между конфликтующими станциями.



Рис. 3.1. Алгоритмы доступа по методу МДКН/ОК

Если какие-либо станции начали передачу в окне столкновений, то по сети распространяются искажённые данные. Это искажение и используется для обнаружения конфликта либо сравнением в передатчике данных, передаваемых в линию (неискажённых) и получаемых из неё (искажённых), либо по появлению постоянной составляющей напряжения в линии, что обусловлено искажением используемого для представления данных манчестерского кода. Обнаружив конфликт, станция должна оповестить об этом партнера по конфликту, посылв дополнительный сигнал затора, после чего станции должны отложить попытки выхода в линию на время t_d . Очевидно, что значения t_d должны быть различными для станций, участвующих в столкновении (конфликте); поэтому t_d – случайная величина.

Среди детерминированных методов преобладают *маркерные методы доступа*. Маркерный метод – метод доступа к среде передачи данных в ЛВС, основанный на передаче полномочий передающей станции с по-мощью специального информационного объекта, называемого маркером. Под полномочием понимается право инициировать определённые действия, динамически предоставляемые объекту, например станции данных в информационной сети.

Применяется ряд разновидностей маркерных методов доступа. Например, в *эстафетном методе* передача маркера выполняется в порядке очередности; в способе *селекторного опроса* (квантированной передачи) сервер опрашивает станции и передаёт полномочие одной из тех станций, которые готовы к передаче. В кольцевых одноранговых сетях широко применяется тактируемый маркерный доступ, при котором маркер циркулирует по кольцу и используется станциями для передачи своих данных.

Оригинальный метод применён в высокоскоростных сетях FDDI, рассматриваемый далее.

3.2. АППАРАТНЫЕ СРЕДСТВА ЛВС

В список сетевого оборудования ЛВС входят моноканалы (другие названия – сегменты, стволы), представляющие собой физические линии передачи данных; сетевые контроллеры (адаптеры, сетевые карты), управляющие доступом к каналу связи; приёмопередатчики, служащие для связи сетевого контроллера с моноканалом; блоки взаимодействия данной сети (или подсети) с другими сетями (подсетями); терминаторы – устройства согласования сопротивлений на концах моноканалов для исключения искажающих отражений сигналов; концентраторы (Hubs) – коммутирующие устройства в сетях звёздной архитектуры; концентраторы оконечных систем – для подключения нескольких ООД; коннекторы – для механического и непосредственного электрического подключения узлов к кабелю.

В качестве линий передачи данных в ЛВС используют коаксиальный кабель, витую (скрученную) пару проводов, волоконно-оптический кабель. Длины используемых отрезков коаксиального кабеля не должны превышать нескольких сотен метров, а у витой пары проводов – десятков метров. При больших расстояниях в среду передачи данных включают формирователи сигналов – повторители для сопряжения отрезков. Волоконно-оптический кабель позволяет существенно увеличить расстояния и скорость передачи данных.

Приёмопередатчик ПП (transiver) – устройство для электрического соединения АКД с линией передачи данных. В состав приёмопередатчика в магистральных ЛВС с методом МДКН/ОК входят:

- *приёмник сигналов* от линии передачи данных; его назначение – усиление информационных сигналов и обнаружение конфликтов путём выделения постоянной составляющей искажённых сигналов и её сопоставления в компараторе с эталонным напряжением;
- *передатчик* от станции в линию; обычно реализуется в виде токового переключателя или балансированной схемы на насыщенных транзисторах с трансформаторным выходом;
- *ответвитель* для подсоединения входов приёмника и выходов передатчика к кабелю; применяется механическое контактирующее устройство, накладываемое на кабель и имеющее винт-иглу, которой прокалывается оплетка кабеля и осуществляется контакт с центральным проводником; игольчатый контакт имеет трансформаторную связь с приёмником и передатчиком сигналов;
- *защита от шума* для отключения ООД от кабеля, если ООД ошибочно генерирует сигналы дольше, чем это предусмотрено.

3.3. СЕТИ ETHERNET

Одной из первых среди ЛВС шинной структуры была создана сеть Ethernet, разработанная фирмой Хегох. В этой сети был применён метод доступа МДКН/ОК. Позднее Ethernet стала основой стандарта IEEE 802/3. Другой вариант шинных ЛВС соответствует стандарту IEEE 802/4, описывающему сеть с эстафетной передачей маркера.

В настоящее время унифицировано несколько вариантов сети Ethernet, различающихся топологией и особенностями физической среды передачи данных.

1. *Вариант Thick Ethernet* (шина «с толстым» кабелем); принятое обозначение варианта 10Base-5, где первый элемент «10» характеризует скорость передачи данных по линии 10 Мбит/с, последний элемент «5» – максимальную длину сегмента (в сотнях метров), т.е. 500 м; другие параметры: максимальное число сегментов 5; максимальное число узлов на одном сегменте 100; минимальное расстояние между узлами 2,5 м. Здесь под сегментом кабеля понимается часть кабеля, используемая в качестве линии передачи данных и имеющая на концах согласующие элементы (терминаторы) для предотвращения отражения сигналов.

2. *Вариант Thin Ethernet* (шина «с тонким» кабелем, *cheapernet*); принятое обозначение 10Base-2: максимальное число сегментов 5; максимальная длина сегмента 185 м; максимальное число узлов на одном сегменте 30; минимальное расстояние между узлами 0,5 м; скорость передачи данных по линии 10 Мбит/с.

3. *Вариант Twisted Pair Ethernet* (топология «звезда»); принятое обозначение 10Base-T, 100Base-T и 1000Base-T; это кабельная сеть с использованием витых пар проводов и концентраторов, называемых также распределителями, или хабами (Hubs). Представление о структуре сети может дать рис. 3.2. В состав сетевого оборудования входят активные (АН) и пассивные (РН) распределители (Active and Passive Hubs), различие между которыми заключается в наличии или отсутствии усиления сигналов и в количестве портов. В одной из разновидностей сети допускаются расстояния между активными распределителями до 600 м в сети 10Base-T и до 70 м в сети 1000Base-T.

4. *Вариант Fiber Optic Ethernet* (шина на основе оптоволоконного кабеля), обозначение 1000Base-F; применяется для соединений «точка-точка», например, для соединения двух конкретных распределителей в кабельной сети. Максимальные длины – в пределах 2...4 км.

5. *Вариант RadioEthernet* (стандарт IEEE 802.11). Среда передачи данных – радиоволны, распространяющиеся в эфире. Структура сети может быть «постоянной» при наличии базовой кабельной сети с точками доступа от узлов по радиоканалам или «временной», когда обмены между узлами происходят только по радиоканалам. Применяется модифицированный метод МДКН/ОК, в котором вместо обнаружения конфликтов используется предотвращение конфликтов.

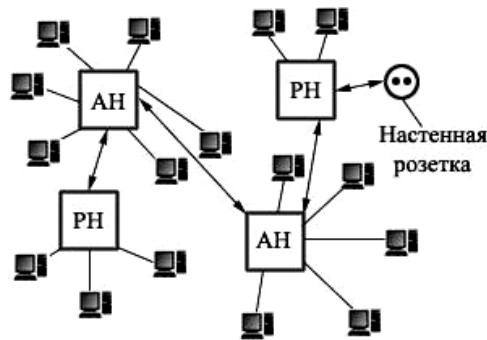


Рис. 3.2. Среда передачи данных на витой паре и концентраторах

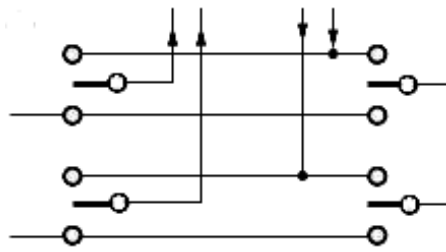
3.4. СЕТЬ TOKEN RING

Из кольцевых ЛВС наиболее распространены сети с передачей маркера по кольцу и среди них: 1) ЛВС типа Token Ring (сеть с таким названием была разработана фирмой IBM и послужила основой для стандарта IEEE 802 /5); 2) сети FDDI (Fiber Distributed Data Interface) на основе ВОЛС.

Топология сети Token Ring показана на рис. 3.3, а. Концентраторы служат для удобства управления сетью, в частности, отключения от кольца неисправных узлов. Схема подключения узлов к кольцу в концентраторах показана на рис. 3.3, б. Для отключения узла достаточно левые переключатели (рис. 3.3, б) поставить в верхнее, а правые переключатели – в нижнее положение (в нормальном режиме положение переключателей противоположное).



а)



б)

Рис. 3.3. Схема сети Token Ring

Функционирование сети основывается на использовании маркера, который циркулирует по кольцу от станции к станции.

Сеть Token Ring рассчитана на меньшие предельные расстояния и число станций, чем Ethernet, но лучше приспособлена к повышенным нагрузкам.

3.5. СЕТЬ FDDI

FDDI (Fiber Distributed Data Interface) – ЛВС кольцевой структуры, использующая ВОЛС и специфический вариант маркерного метода доступа.

В основном варианте сети применено двойное кольцо на ВОЛС. Используются волны длиной 1300 нм.

Два кольца ВОЛС используются одновременно. Станции можно подключать к одному из колец или к обоим сразу. Использование конкретным узлом обоих колец позволяет для этого узла иметь суммарную пропускную способность в 2000 Мбит/с.

В FDDI используются оригинальные код и метод доступа. Применяется код типа NRZ (без возвращения к нулю), в котором изменение полярности в очередном такте времени воспринимается как 1, отсутствие изменения полярности как 0. Чтобы код был самосинхронизирующимся, после каждых четырёх битов передатчик вырабатывает синхронизирующий перепад.

В соответствии с методом FDDI по кольцу циркулирует пакет, состоящий из маркера и информационных кадров. Любая станция, готовая к передаче, распознав проходящий через неё пакет, вписывает свой кадр в конец пакета. Она же ликвидирует его после того, как кадр вернётся к ней после оборота по кольцу и при условии, что он был воспринят получателем. Если обмен происходит без сбоев, то кадр, возвращающийся к станции-отправителю, оказывается в пакете уже первым, так как все предшествующие кадры должны быть ликвидированы раньше.

Сеть FDDI обычно используется как объединяющая в единую сеть много отдельных подсетей ЛВС. Например, при организации информационной системы крупного предприятия целесообразно иметь ЛВС типа Ethernet или Token Ring в помещениях отдельных проектных подразделений, а связь между подразделениями осуществлять через сеть FDDI.

4. СЕТЕВОЙ И ТРАНСПОРТНЫЙ УРОВНИ

4.1. ТРАНСПОРТНЫЕ И СЕТЕВЫЕ ПРОТОКОЛЫ

В территориальных и корпоративных сетях со сложной разветвлённой структурой для передачи сообщений от отправителя к адресату имеется много альтернативных маршрутов. Эти маршруты, как правило, включают не только конечные узлы отправителя и получателя, но и ряд промежуточных узлов и сетей.

Проблема обеспечения передачи информации между сетями, т.е. проблема обеспечения взаимодействия различных сетей в составе интегрированной сети, в англоязычной литературе носит название *Internetworking*. Это взаимодействие выражено функциями транспортного и сетевого уровней в семиуровневой модели ISO.

Функции *транспортного* уровня реализуются в конечных узлах. К ним относятся:

- разделение пакета на дейтаграммы, если сеть работает без установления соединения;
- сборка сообщений из дейтаграмм в узле-получателе;
- обеспечение заданного уровня услуг, включающих заказ времени доставки, типа канала связи, возможности сжатия данных с частичной потерей информации (как, например в алгоритме JPEG);
- управление сквозными соединениями в сети с помощью специальных команд запроса соединения, разъединения, передачи, приёма, регистрации и др.

Назначение *сетевых* протоколов – приспособление пакетов к особенностям промежуточных сетей и выбор направления передачи пакетов (маршрутизация). В список основных функций входят:

- формирование пакетов с учётом требований промежуточных сетей (дополнение пакетов транспортного уровня обрамлением, включающим флаги, сетевые адреса получателя и отправителя, служебную информацию);
- управление потоками;
- маршрутизация;
- обнаружение неисправностей;
- ликвидация «заблудившихся» дейтаграмм и т.п.

Наиболее широко используемыми протоколами на сетевом уровне являются протоколы IP (Internet Protocol), X.25, IPX (Internet Packet Exchange) и на транспортном уровне TCP (Transmission Control Protocol) и SPX (Sequence Packet Exchange).

Последние входят в систему протоколов TCP/IP и SPX/IPX, соответственно. Протоколы TCP/IP первоначально были разработаны для сети ARPANET, а затем на их основе стала развиваться сеть Internet. Протоколы SPX/IPX разработаны и применяются фирмой Novell для сетей Novell Netware, объединяющих персональные ЭВМ. Протоколы X.25 разработаны ITU и включают части для физического, канального и сетевого уровней.

4.2. УПРАВЛЕНИЕ ПОТОКАМИ ДАННЫХ В СЕТЯХ

Управление потоками данных – это одна из функций сетевого уровня, включающая управление нагрузками и борьбу с блокировками. Различают несколько уровней управления.

Межузловое управление связано с распределением буферной памяти в промежуточных узлах (выделением каждому направлению определённого числа буферов), сводящееся к ограничению длин канальных очередей.

Управление «вход-выход» направлено на предотвращение блокировок. Реализуется указанием в первом пакете сообщения его длины, что позволяет приёмному узлу прогнозировать заполнение памяти и запрещать приём дейтаграмм определённых сообщений, если прогнозируется блокировка памяти.

Управление внешними потоками (доступом) реализуется путём предо-ставления приоритета в передаче внутренним потокам перед внешними, ограничением числа пакетов в сети (пакет принимается, если у узла есть соответствующее разрешение), посылкой предупредительных пакетов-заглушек в адрес источника, от которого идут пакеты в перегруженную линию связи.

4.3. МАРШРУТИЗАЦИЯ

Цель маршрутизации – доставка пакетов по назначению с максимизацией эффективности. Чаще всего эффективность выражена взвешенной суммой времён доставки сообщений при ограничении снизу на вероятность доставки. Маршрутизация сводится к определению направлений движения пакетов в маршрутизаторах. Выбор одного из возможных в маршрутизаторе направлений зависит от текущей топологии сети (она может меняться хотя бы из-за временного выхода некоторых узлов из строя), длин очередей в узлах коммутации, интенсивности входных потоков и т.п.

Алгоритмы маршрутизации включают процедуры:

- измерение и оценивание параметров сети;
- принятие решения о рассылке служебной информации;
- расчёт таблиц маршрутизации (ТМ);
- реализация принятых маршрутных решений.

В зависимости от того, используется ли при выборе направления информация о состоянии только данного узла или всей сети, различают алгоритмы *изолированные* и *глобальные*. Если ТМ реагируют на изменения состояния сети, то алгоритм *адаптивный*, иначе *фиксированный (статический)*, а при редких корректировках – *квазистатический*. В статических маршрутизаторах изменения в ТМ вносит администратор сети.

Простейший алгоритм – изолированный, статический. *Алгоритм кратчайшей очереди* в отличие от простейшего является адаптивным, пакет посылается по направлению, в котором наименьшая очередь в данном узле. *Лавинный алгоритм* – многопутевой, основан на рассылке копий пакета по всем направлениям, пакеты сбрасываются, если в данном узле другая копия уже проходила. Очевидно, что лавинный алгоритм обеспечивает надёжную доставку, но порождает значительный трафик и потому используется только для отдельных пакетов большой ценности.

Наиболее широко используемые протоколы маршрутизации – RIP (Routing Information Protocol) и OSPF (Open Shortest Path First). Метод RIP иначе называется методом рельефов. Он основан на *алгоритме Беллмана-Форда* и используется преимущественно на нижних уровнях иерархии. OSPF – алгоритм динамической маршрутизации, в котором информация о любом изменении в сети рассылается лавинообразно.

Алгоритм Беллмана-Форда относится к алгоритмам DVA (Distance Vector Algorithms). В DVA *рельеф* $R_a(d)$ – это оценка кратчайшего пути от узла a к узлу d . Оценка (условно назовем её расстоянием) может выражаться временем доставки, надёжностью доставки или числом узлов коммутации (измерение в хопах) на данном маршруте. В ТМ узла, a каждому из остальных узлов отводится одна строка со следующей информацией:

- узел назначения;
- длина кратчайшего пути;
- номер N ближайшего узла, соответствующего кратчайшему пути;
- список рельефов от a к d через каждый из смежных узлов.

Например, на рис. 4.1 в узле a строка для d выглядит как

$$d R_a(d) N(d) = j R_{aj}(d) R_{ak}(d).$$

Пусть изменилась задержка $R_{ak}(d)$ причём так, что $R_{ak}(d)$ стало меньше, чем $R_{aj}(d)$. Тогда в строке d таблицы маршрутизации узла a корректируется $R_a(d)$, $N(d)$ изменяется на k и, кроме того, всем соседям узла a посылается сообщение об изменённом $R_a(d)$. Например, в некотором соседнем узле l при этом будет изменено значение $R_{la}(d) = R_{la}(d) + R_l(a)$. Мы видим, что возникает итерационный процесс корректировки маршрутной информации в узлах маршрутизации.

Хотя алгоритм Беллмана-Форда сходится медленно, для сетей сравнительно небольших масштабов он вполне приемлем. В больших сетях лучше себя зарекомендовал алгоритм OSPF. Он основан на использовании в каждом маршрутизаторе информации о состоянии всей сети. В основе OSPF лежит алгоритм Дейкстры поиска кратчайшего пути в графах. При этом сеть моделируется графом, в котором узлы соответствуют маршрутизаторам, а рёбра – каналам связи. Веса рёбер – оценки (расстояния) между инцидентными узлами. Рассмотрим итерационный алгоритм Дейкстры применительно к формированию маршрутной таблицы в узле a графа, показанного на рис. 4.2 (числа показывают веса рёбер).

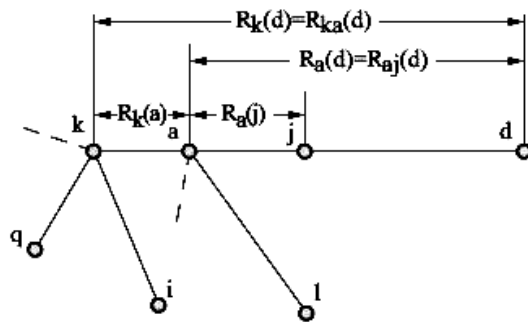


Рис. 4.1. Пояснение к методу маршрутизации RIP

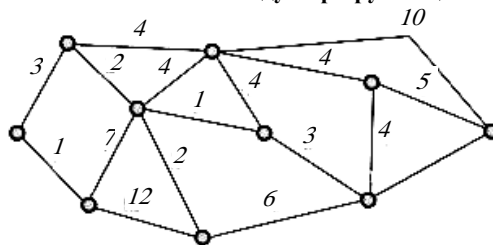


Рис. 4.2. Сеть для примера маршрутизации по алгоритму OSPF

Обозначим кратчайшее расстояние от a к l через R_l . Разделим узлы на три группы: 1) перманентные, для которых R_i уже рассчитано; 2) пробные, для которых получена некоторая промежуточная оценка R_i , возможно не окончательная; 3) пассивные, ещё не вовлечённые в итерационный процесс. Итерационный процесс начинается с отнесения узла a к группе перманентных. Далее определяются узлы, смежные с узлом a . Это узлы b и c , которые включаются в группу пробных. Включение в группу пробных отмечается указанием в клетке таблицы рядом с оценкой расстояния пробного узла также имени узла, включаемого на этом шаге в число перманентных. Так, для узлов b и c определяются расстояния $R_b = 3$, $R_c = 1$ и для них в таблице отмечается узел a . На следующем шаге узел с минимальной оценкой (в примере это узел c) включается в группу перманентных, а узлы, смежные с узлом c , – в группу пробных, для них оцениваются расстояния $R_d = 8$ и $R_f = 13$ и они помечаются символом c . Теперь среди пробных узлов минимальную оценку имеет узел b , он включается в группу перманентных узлов, узел e – в группу пробных и для всех пробных узлов, смежных с b , рассчитываются оценки. Это, в частности, приводит к уменьшению оценки узла d с 8 на 5. Акт уменьшения фиксируется (в таблице это отражено, во-первых, подчеркиванием, а во-вторых, заменой у узла d метки c на b). Если же новая оценка оказывается больше прежней, то она игнорируется. Этот процесс продолжается, пока все узлы не окажутся в группе перманентных. Теперь виден кратчайший путь от узла a к любому другому узлу X или, что то же самое, от X к a . Это последовательность конечных отметок в строках таблицы, начиная с последнего узла X . Так, для узла $X = n$ имеем в строке n отметку h , в строке h – отметку g , в строке g – отметку d и т.д. и окончательно кратчайший путь есть $a-b-d-g-h-n$.

4.4. ТРАНСПОРТНЫЙ ПРОТОКОЛ TCP В СТЕКЕ ПРОТОКОЛОВ TCP/IP

Эти протоколы берут своё начало от одной из первых территориальных сетей ARPANET. Они получили широкое распространение благодаря реализации в ОС Unix и в сети Internet и в настоящее время оформлены в виде стандартов RFC (Requests For Comments) организацией IETF (Internet Engineering Task Force).

TCP/IP – пятиуровневые протоколы, но основными среди них, давшими название всей совокупности, являются протоколы сетевого (IP – Internet Protocol) и транспортного (TCP – Transport Control Protocol) уровней.

TCP – дуплексный транспортный протокол с установлением соединения. Его функции: упаковка и распаковка пакетов на концах транспортного соединения; установление виртуального канала путём обмена запросом и согласием на соединение; управление потоком – получатель при подтверждении правильности передачи сообщает размер окна, т.е. диапазон номеров пакетов, которые получатель готов принять; помещение срочных данных между специальными указателями, т.е. возможность управлять скоростью передачи.

Схема установления соединения в одноранговых сетях такова: инициатор соединения обращается к своей ОС, которая в ответ выдаёт номер протокольного порта и посылает сегмент получателю. Тот должен подтвердить получение запроса и послать свой сегмент-запрос на создание обратного соединения (так как соединение дуплексное). Инициатор должен подтвердить создание обратного соединения. Получается трёхшаговая процедура установления соединения. Во время этих обменов партнеры сообщают номера байтов в потоках данных, с которых начинаются сообщения. На противоположной стороне счётчики устанавливаются в состояние на единицу больше, чем и обеспечивается механизм синхронизации в дейтаграммной передаче, реализуемой на сетевом уровне. После установления соединения начинается обмен. При этом номера протокольных портов включаются в заголовок пакета. Каждое соединение (socket) получает свой идентификатор ISN. Разъединение происходит в обратном порядке.

Схема установления соединения в сетях «клиент-сервер» аналогична и включает послышку клиентом запроса на соединение с указанием адреса сервера, тайм-аута (времени жизни), уровня секретности. Можно сразу же поместить в запрос данные. Если сервер готов к связи, он отвечает командой согласия, в которой назначает номер соединения. Далее командой SEND посылаются данные, а командой DELIVER подтверждается их получение. Разъединение выполняется обменом командами CLOSE и CLOSING.

Каждый байт сообщения получает уникальный порядковый номер. Отсюда вытекает одно из ограничений на максимально допустимую в протоколе TCP/IP пропускную способность. Это ограничение составляет $(2^{32} \text{ байта}) / (\text{время жизни дейтаграммы})$, так как для конкретного соединения в сети не должно одновременно существовать более одного байта с одним и тем же номером.

Ещё более жёсткое ограничение возникает из-за представления размера окна всего 16-ю битами. Это ограничение заключается в том, что за время T_v прохождения пакета от отправителя к получателю и обратно в сеть может быть направлено не более 2^{16} информационных единиц конкретного сообщения. Поскольку обычно такой единицей является байт, то имеем $(2^{16} \cdot 8 \text{ бит}) / T_v$.

В TCP повторная передача пакета происходит, если в течение оговорённого интервала времени Tm (тайм-аута) не пришло положительное подтверждение. Следовательно, не нужно посылать отрицательные квитанции. Обычно $Tm = 2t$, где t – некоторая оценка времени прохождения пакета туда и обратно. Это время периодически корректируется по результату измерения T_v , а именно

$$t = 0,9t + 0,1 T_v$$

Попытки повторных передач пакета не могут продолжаться бесконечно, и при превышении интервала времени, устанавливаемого в пределах 0,5...2,0 мин, соединение разрывается.

Размер окна регулируется следующим образом. Если сразу же после установления соединения выбрать завышенный размер окна, что означает разрешение посылки пакетов с высокой интенсивностью, то велика вероятность появления перегрузки определённых участков сети. Поэтому используется алгоритм так называемого медленного старта. Сначала посылается один пакет и после подтверждения его приёма окно увеличивается на единицу, т.е. посылаются два пакета. Если вновь положительное подтверждение (потерь пакетов нет), то посылаются уже четыре пакета и т.д. Скорость растёт, пока пакеты проходят успешно. При потере пакета или при приходе от протокола управления сигнала о перегрузке размер окна уменьшается и далее опять возобновляется процедура линейного роста размера окна. Медленный старт снижает информационную скорость, особенно при пересылке коротких пакетов, поэтому стараются применять те или иные приёмы его улучшения.

4.5. КОММУТАЦИОННОЕ ОБОРУДОВАНИЕ

Существует несколько видов коммуникационного оборудования.

Маршрутизатор (router) – блок взаимодействия, служащий для выбора маршрута передачи данных в корпоративных и территориальных сетях. С помощью маршрутизаторов могут согласовываться не только каналные протоколы, как это имеет место при применении мостов, но и сетевые протоколы. Маршрутизаторы содержат таблицы и протоколы маршрутизации в отличие от других узлов (узлы, имеющие адреса и, следовательно, доступные по сети, называют хостами), которые могут содержать лишь локальные таблицы соответствия IP-адресов физическим адресам сетевых контроллеров в локальной сети.

Шлюз (gateway – межсетевой преобразователь) – блок взаимодействия, служащий для соединения информационных сетей различной архитектуры и с неодинаковыми протоколами. В шлюзах предусматривается согласование протоколов всех семи уровней ЭМВОС. Примерами шлюзов могут быть устройства, соединяющие ЛВС типа Ethernet с сетью SNA, используемой для связи больших машин фирмы IBM. Часто под шлюзом понимают сервер, имеющий единственный внешний канал передачи данных.

Концентраторы предназначены для объединения в сеть многих узлов. Концентраторы могут быть пассивными или активными, в последнем случае имеются усилители-формирователи подводимых сигналов. Однако такие концентраторы создают общую среду передачи данных без разделения трафика.

Коммутаторы в отличие от концентраторов предназначены для объединения в сеть многих узлов или подсетей с возможностью создания одновременно многих соединений. Они называются также переключателями (свитчами – switches). Коммутаторы используются также для связи нескольких ЛВС с территориальной сетью. Один коммутатор может объединять несколько как однотипных, так и разнотипных ЛВС.

Использование коммутаторов вместо маршрутизаторов (там, где это возможно) позволяет существенно повысить пропускную способность сети. Коммутатор работает с локальными MAC-адресами, в нём имеется таблица соответствия MAC-адресов и портов. Кроме того, между разными портами коммутатора образуется несколько соединений, по которым пакеты могут передаваться одновременно. В то же время маршрутизатор оперирует IP-адресами и таблицами маршрутизации и выполняет сложные алгоритмы маршрутизации.

Возможны коммутация «на лету» (сквозная коммутация – out-trough), когда передача пакета начинается сразу после расшифровки заголовка, и с полным получением пакета (промежуточная буферизация – store-and-forward). Первый способ применяют в небольших сетях, второй – в магистральных коммутаторах. Сквозная коммутация позволяет обойтись малым объёмом буфера, но не даёт возможности контролировать безошибочность передачи данных.

Виртуальная ЛВС (ВЛВС) – это локальная сеть, в которой узлы группируются не по территориальному, а по функциональному признаку. Для этого каждая подсеть в ВЛВС получает свой идентификатор, каждому идентификатору соответствуют определённые порты коммутаторов сети.

К блокам взаимодействия относят также модемы, *мультиплексоры* и *демультиплексоры* – устройства для преобразования сообщений в кадры TDM (временное мультиплексирование) и обратно.

4.6. СЕТЕВОЙ ПРОТОКОЛ IP В СТЕКЕ ПРОТОКОЛОВ TCP/IP

IP – дейтаграммный сетевой протокол без установления соединения. Его функции: фрагментация и сборка пакетов при прохождении через промежуточные сети, имеющие другие протоколы; маршрутизация; проверка контрольной суммы заголовка пакета (правильность передачи всего пакета проверяется на транспортном уровне, т.е. с помощью TCP, в оконечном узле); управление потоком – сброс дейтаграмм при превышении заданного времени жизни.

Всего в сети одновременно может быть $2^{16} = 65$ тысяч дейтаграмм сообщения с разными идентификаторами, т.е. за отрезок времени, равный времени жизни дейтаграммы, может быть передано не более 2^{16} дейтаграмм. Это один из факторов, ограничивающих пропускную способность сетей с протоколом IP. Действительно, при времени жизни в 120 с имеем предельную скорость $2^{16} / 120 = 546$ дейтаграмм в секунду, что при размере дейтаграммы до 65 тысяч байт даёт ограничение скорости приблизительно в 300 Мбит/с (такое же значение одного из ограничений предельной скорости получено выше и для протокола TCP).

Время жизни может измеряться как в единицах времени T , так и в хопах P (числом пройденных маршрутизаторов). В первом случае контроль ведётся по записанному в заголовке значению T , которое уменьшается на единицу каждую секунду. Во втором случае каждый маршрутизатор уменьшает число P , записанное в поле «Время жизни», на единицу. При $T = 0$ или при $P = 0$ дейтаграмма сбрасывается.

Поле «Тип протокола» определяет структуру данных в дейтаграмме. Примерами протоколов могут служить UDP, SNA, IGP и т.п.

Поле «Опции» в настоящее время рассматривается как резервное.

4.7. ДРУГИЕ ПРОТОКОЛЫ В СТЕКЕ TCP/IP

В состав протокола IP входит ряд частных протоколов. Среди них протоколы ARP, IGP, EGP, относящиеся к маршрутизации на разных иерархических уровнях в архитектуре сети. На одном уровне с IP находится протокол управления ICMP (Internet Control Message Protocol).

Протокол ARP (Address Resolution Protocol) относится к связям «хост-хост» или «хост-шлюз» в конкретной подсети. Он использует локальные таблицы маршрутизации – ARP-таблицы, устанавливающие соответствие IP-адресов с NPA (Network Point of Attachment) адресами серверов доступа в соответствующих подсетях. В подсетях не нужно рассчитывать кратчайший путь и определять маршрут в разветвлённой сети, что, естественно, ускоряет доставку. ARP-таблицы имеются в каждом узле. Если в таблице отправителя нет строки для IP-адреса получателя, то отправитель сначала посылает широковещательный запрос. Если некоторый узел имеет этот IP-адрес, он откликается своим NPA, и отправитель пополняет свою таблицу и отсылает пакет. Иначе отправка пакета произойдёт на внешний порт сети.

Протокол IGP (Interior Gateway Pr.) предназначен для управления маршрутизацией в некотором домене (автономной сети – AS), т.е. он определяет маршруты между внутренними сетями домена. Другими словами, в AS имеется (или может быть получена) информация о путях ко всем сетям домена, и протокол IGP доставляет дейтаграмму в нужную подсеть в соответствии с алгоритмом маршрутизации RIP или OSPF.

Протокол EGP (Exterior Gateway Pr.) относится к корневой сети и предназначен для управления маршрутизацией между внешними шлюзами и пограничными маршрутизаторами доменов.

В TCP/IP входит также протокол UDP (User Datagram Protocol) – транспортный протокол без установления соединения, он значительно проще TCP, но используется чаще всего для сообщений, умещающихся в один пакет. После оформления UDP-пакета он передаётся с помощью средств IP к адресату, который по заголовку IP-пакета определяет тип протокола и передаёт пакет не агенту TCP, а агенту UDP. Агент определяет номер порта и ставит пакет в очередь к этому порту. В UDP служебная часть дейтаграммы короче, чем в TCP (8 байт вместо 20), не требуется предварительного установления соединения или подтверждения правильности передачи, как это делается в TCP, что и обеспечивает большую скорость за счёт снижения надёжности доставки.

Структура UDP-дейтаграммы (в скобках указано число битов):

Часто считают, что в состав TCP/IP входят также протоколы высоких уровней такие, как:

- SMTP (Simple Mail Transport Protocol) – почтовый протокол, который по классификации ISO можно было бы отнести к прикладному уровню;
- FTP (File Transfer Protocol) – протокол с функциями представительного уровня;
- Telnet – протокол с функциями сеансового уровня.

Протоколы TCP/IP являются основными протоколами сети Internet, они поддерживаются операционными системами Unix и Windows NT.

На нижних уровнях в TCP/IP используется протокол IEEE 802.X или X.25.

4.8. АДРЕСАЦИЯ В TCP/IP (В INTERNET)

Различают два типа адресов. На канальном уровне используют адреса, называемые *физическими*. Это шестибайтовые адреса сетевых плат, присваиваемые изготовителем контроллеров (каждый изготовитель вместе с лицензией на изготовление получает уникальный диапазон адресов). На сетевом уровне используют сетевые адреса, иначе называемые *виртуальными*, или *логическими*. Эти адреса имеют иерархическую структуру, для них существуют цифровое и буквенное выражения.

Узлы в Internet имеют адрес и имя. Адрес – уникальная совокупность чисел: адреса сети и компьютера (хоста – узла в сети), которая указывает их местоположение. Имя характеризует пользователя. Оно составляется в соответствии с доменной

системой имен. Соответствие между IP-адресом и IP-именем хоста устанавливается специальной *службой директорий*. В Internet это DNS (Domain Name Service), в ISO – стандарт X.500.

IP-имя, называемое также *доменным именем*, – удобное для человека название узла или сети. Имя отражает иерархическое построение глобальных сетей и потому состоит из нескольких частей (аналогично обычным почтовым адресам). Корень иерархии обозначает либо страну, либо отрасль знаний, например: ru – Россия, us – США, de – Германия, uk – Великобритания, edu – наука и образование, com – коммерческие организации, org – некоммерческие организации, gov – правительственные организации, mil – военные ведомства, net – служба поддержки Internet и т.д. Корень занимает в IP-имени правую позицию, левее записываются локальные части адреса и, наконец, перед символом @ указывается имя почтового ящика пользователя. Так, запись norenkov@rk6.bmstu.ru расшифровывается, как пользователь norenkov в подразделении rk6 организации bmstu в стране ru. В 1997 г. число используемых доменных имён в сети Internet превысило один миллион.

IP-адрес – 32-битовое слово, записываемое в виде четырёх частей (побайтно), разделённых точками. Каждая подсеть и узел в подсети получают свои номера, причём для сети (подсети) может использоваться от одного до трёх старших байтов, а оставшиеся байты – для номера узла. Какая часть IP-адреса относится к сети, определяется её маской, выделяющей соответствующие биты в IP-адресе. Например, для некоторой сети маска может быть 255.0.0.0, а для её подсети – 255.255.0.0 и т.д. Тем самым описывается иерархия сетей.

Номера при включении нового хоста выдаёт организация, предоставляющая телекоммуникационные услуги и называемая *провайдером*. Провайдер, в частности, обеспечивает включение IP-адреса и соответствующего ему IP-имени в сервер службы адресов DNS. Это означает запись данных о хосте в DIB (Directory Information Base) локального узла DNS.

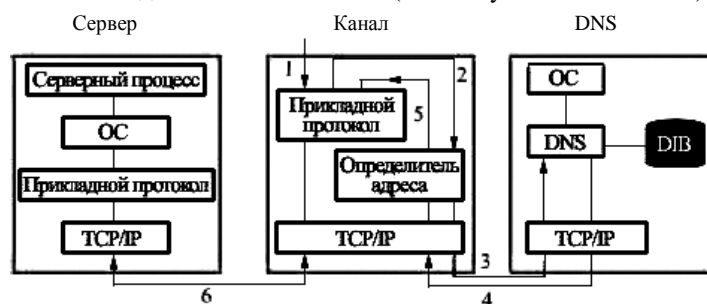


Рис. 4.3. Определение IP-адреса с помощью DNS-сервера

При маршрутизации имя переводится в адрес с помощью серверов DNS (Domain Name Service). Поскольку маршрутизация в сети осуществляется по IP-адресам, то перевод указанного пользователем IP-имени в IP-адрес с помощью DNS обязателен.

Сценарий работы с DNS иллюстрирует рис. 4.3.

Маршрутизация в Internet организована по иерархическому принципу, имеются уровни ЛВС (корпоративных сетей), маршрутных доменов (RD – Routing Domains), в каждом из которых используются единые протоколы и алгоритмы маршрутизации, административных доменов (AD), каждый из которых соответствует некоторой ассоциации и имеет единое управляющее начало. В RD имеются внешние маршрутизаторы для связи с другими RD или с AD. Обращение из некоторого узла к Internet (например, из wwwcdl.bmstu.ru по адресу http://www.eevl.ac.uk) происходит к местному серверу (bmstu), и если там сведений об адресе назначения нет, то происходит переход к серверу следующего, более высокого уровня (ru) и далее по иерархии вниз до получения IP-адреса хоста назначения. В местном DNS сервере могут быть сведения об IP-адресах хостов из удалённых доменов, если к ним происходят достаточно частые обращения из данного домена.

4.9. ПРОТОКОЛЫ УПРАВЛЕНИЯ В СТЕКЕ TCP/IP

Рост сложности сетей повышает значимость и сложность средств управления сетью.

Среди протоколов управления различают протоколы, реализующие управляющие функции сетевого уровня, и протоколы мониторинга за состоянием сети, относящиеся к более высоким уровням. В сетях TCP/IP роль первых из них выполняет протокол ICMP, роль вторых – протокол SNMP (Simple Network Management Protocol).

Основные функции *ICMP*:

- оповещение отправителя с чрезмерным трафиком о необходимости уменьшить интенсивность посылки пакетов; при перегрузке адресат (или промежуточный узел) посылает ICMP-пакеты, указывающие о необходимости сокращения интенсивности входных потоков;
- передача откликов (квитанций) на успешно переданные пакеты;
- контроль времени жизни T дейтаграмм и их ликвидация при превышении T или по причине искажения данных в заголовке;
- оповещение отправителя о недостижимости адресата; Отправление ICMP-пакета с сообщением о невозможности достичь адресата осуществляет маршрутизатор;
- формирование и посылка временных меток (измерение задержки) для контроля T_v – времени доставки пакетов, что нужно для «оконного» управления. Например, время доставки T_v определяется следующим образом. Отправитель формирует ICMP-запрос с временной меткой и отправляет пакет. Получатель меняет адреса местами и отправляет пакет обратно. Отправитель сравнивает метку с текущим временем и тем самым определяет T_v .

ICMP-пакеты вкладываются в IP-дейтаграммы при доставке.

Основные функции протоколов мониторинга заключаются в сборе информации о состоянии сети, в предоставлении этой информации нужным лицам путём посылки её на соответствующие узлы, в возможном автоматическом принятии необходимых управляющих мер.

Собственно собираемая информация о состоянии сети хранится в базе данных под названием MIB (Management Information Base). Примеры данных в MIB: статистика по числу пакетов и байтов, отправленных или полученных правильно или с ошибками, длины очередей, максимальное число соединений и др.

Протокол SNMP относится к прикладному уровню в стеке протоколов TCP/IP. Он работает по системе «менеджер-агент». Менеджер (серверная программа SNMP) посылает запросы агентам, агенты (т.е. программы SNMP объектов управления) устанавливаются в контролируемых узлах, они собирают информацию (например, о загрузке, очередях, временах совершения событий), и передают её серверу для принятия нужных мер. В общем случае агентам можно поручить и обработку событий, и автоматическое реагирование на них. Для этого в агентах имеются триггеры, фиксирующие наступление событий, и средства их обработки. Команды SNMP могут запрашивать значения объектов MIB, посылать ответы, менять значения параметров.

Для посылки команд SNMP используется транспортный протокол UDP.

Одной из проблем управления по SNMP является защита агентов и менеджеров от ложных команд и ответов, которые могут дезорганизовать работу сети. Используется шифрование сообщений, но это снижает скорость реакции сети на происходящие события.

Расширением SNMP являются протоколы RMON (Remote Monitoring) для сетей Ethernet и Token Ring и RMON2 для сетевого уровня. Преимущество RMON заключается в меньшем трафике, так как здесь агенты более самостоятельны и сами выполняют часть необходимых управляющих воздействий на состоянии контролируемых ими узлов.

На базе протокола SNMP разработан ряд мощных средств управления, примерами которых могут служить продукт ManageWISE фирмы Novell или система UnicenterTNG фирмы Computer Associates. С их помощью администратор сети может: 1) строить 2D изображение топологии сети, причём на разных иерархических уровнях, перемещаясь от региональных масштабов до подсетей ЛВС (при интерактивной работе); 2) разделять сеть на домены управления по функциональным, географическим или другим принципам с установлением своей политики управления в каждом домене; 3) разрабатывать нестандартные агенты с помощью имеющихся инструментальных средств.

Дальнейшее развитие подобных систем может идти в направлении связи сетевых ресурсов с проектными или бизнес-процедурами и сетевых событий с событиями в процессе проектирования или управления предприятиями. Тогда система управления сетью станет комплексной системой управления процессами проектирования и управления предприятием.

4.10. ФУНКЦИИ И ХАРАКТЕРИСТИКИ СЕТЕВЫХ ОПЕРАЦИОННЫХ СИСТЕМ

Различают операционные системы (ОС) со встроенными сетевыми функциями и оболочки над локальными ОС. По другому признаку классификации различают сетевые ОС одноранговые и функционально несимметричные (для систем «клиент/сервер»).

Основные функции сетевой ОС:

- управление каталогами и файлами;
- управление ресурсами;
- коммуникационные функции;
- защита от несанкционированного доступа;
- обеспечение отказоустойчивости;
- управление сетью.

Управление каталогами и файлами является одной из первоочередных функций сетевой операционной системы, обслуживаемых специальной сетевой файловой подсистемой. Пользователь получает от этой подсистемы возможность обращаться к файлам, физически расположенным в сервере или в другой станции данных, применяя привычные для локальной работы языковые средства. При обмене файлами должен быть обеспечен необходимый уровень конфиденциальности обмена (секретности данных).

Управление ресурсами включает запросы и предоставление ресурсов.

Коммуникационные функции обеспечивают адресацию, буферизацию, маршрутизацию.

Защита от несанкционированного доступа возможна на любом из следующих уровней: ограничение доступа в определённое время, и (или) для определённых станций, и (или) определённое число раз; ограничение совокупности доступных конкретному пользователю директорий; ограничение для конкретного пользователя списка возможных действий (например, только чтение файлов); пометка файлов символами типа «только чтение», «скрытность при просмотре списка файлов».

Отказоустойчивость определяется наличием в сети автономного источника питания, отображением или дублированием информации в дисковых накопителях. Отображение заключается в хранении двух копий данных на двух дисках, подключенных к одному контроллеру, а дублирование означает подключение каждого из этих двух дисков к разным контроллерам. Сетевая ОС, реализующая дублирование дисков, обеспечивает более высокий уровень отказоустойчивости.

Дальнейшее повышение отказоустойчивости связано с дублированием серверов.

Чем сложнее сеть, тем острее встают вопросы *управления сетью*. Основные функции управления сетью реализуются в ПО, поддерживающем протоколы управления такие, как ICMP и SNMP или протокол ISO для семиуровневой модели СМIP

(Common Management Information Protocol). Как рассмотрено выше, это ПО представлено менеджерами и агентами. Менеджер – прикладная программа, выдающая сетевые команды. Агенты доводят эти команды до исполнительных устройств и сигнализируют о событиях в состоянии устройств, они следят за трафиком и фиксируют аномалии, помогают восстановлению информации после сбоев, борются с вирусами и т.п.

В сетевых ОС обычно выделяют ядро, реализующее большинство из перечисленных функций и ряд дополнительных программ (служб), ориентированных на реализацию протоколов верхних уровней, организацию распределённых вычислений и т.п. К сетевому программному обеспечению относятся также драйверы сетевых плат, различные для разных типов ЛВС (Ethernet, TR, AppleTalk и др.). Но и внутри одного типа ЛВС имеется много плат с разными характеристиками интеллектуальности, скорости, объема буферной памяти.

В настоящее время (2008) выбор среди ОС происходит преимущественно между тремя основными системами – UNIX, Windows NT, Novell Netware.

Областью применения ОС UNIX остаются крупные TCP/IP корпоративные сети. Отличительные свойства UNIX – высокая надёжность, возможность лёгкого масштабирования сети.

4.11. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ В СЕТЯХ

Проблема информационной безопасности (ИБ) выходит за рамки сетевой ОС. Назначение систем ИБ сводится к защите от несанкционированного доступа и модификации информации, восстановлению после разрушений информации. Функции систем ИБ: аутентификация, разграничение доступа, защита на сетевом уровне.

Разграничение доступа должно обеспечиваться на нескольких уровнях. На внешнем уровне устанавливаются права доступа извне и выхода изнутри корпоративной сети. На сетевом, системном и прикладном уровнях регламентируются права доступа к сетевым информационным ресурсам, ресурсам ОС и к пользовательским данным, соответственно. Другая модель устанавливает уровни входа в систему, доступа к БД, доступа к приложениям. Права доступа часто выражаются трёхразрядным восьмеричным кодом *ABC*, в котором *A* – права владельца, *B* – членов группы, *C* – остальных пользователей, а три бита выражают право чтения, записи и исполнения, соответственно.

Между общедоступными и секретными объектами в сети (между общедоступными и частными сетями) можно установить специальное программное обеспечение, называемое брандмауэром (или firewall), которое либо запрещает выполнение определённых действий на сервере, либо фильтрует пакеты, разрешая проход только от оговоренных узлов.

Борьба с перехватом сообщений на сетевом уровне – шифрование при передаче через канал (криптография). Разработан стандарт шифрования DES (Data Encryption Standard).

Различают симметричную и асимметричную схемы шифрования.

В симметричных схемах секретный ключ должен быть известен как отправителю, так и получателю. Это затрудняет смену ключей, полезность которой очевидна.

В асимметричных схемах шифрование производится открытым ключом, а дешифрование – секретным ключом, известным только получателю. Случайно подобрать секретный ключ злоумышленник не может, так как это требует громадного перебора вариантов.

В настоящее время всё большее распространение получает комбинация симметричных и асимметричных схем. При этом сообщение кодируется закрытым ключом *A* по симметричной схеме, но сам ключ *A* для каждого сообщения новый и передаётся в закодированном по асимметричной схеме виде вместе с сообщением. Получатель декодирует сначала ключ *A* своим закрытым ключом *B*, а затем и всё сообщение ключом *A*. Такая комбинация выгодна, во-первых, тем, что труднее взломать защиту, во-вторых, получатель быстрее дешифрует сообщения, так как алгоритмы симметричного дешифрования заметно более экономичны.

Одним из применений шифрования является электронная подпись, предназначенная для удостоверения подлинности документа, пересылаемого по сети. Документ (чаще его аннотация) перед отправкой шифруется секретным ключом отправителя, а дешифруется открытым ключом получателя.

Часть II

КОМПЬЮТЕРНАЯ ГРАФИКА

1. АЛГОРИТМ БРЕЗЕНХЕМА ДЛЯ ПЕРОБРАЗОВАНИЯ ОТРЕЗКОВ ПРЯМЫХ ИЗ ВЕКТОРНОЙ ФОРМЫ В РАСТРОВУЮ

Любой алгоритм развёртки отрезка из векторной формы в растровую сводится к вычислению координат пикселей, лежащих вблизи отрезка на двумерной растровой сетке. При решении этой задачи будем предполагать, что начальная и конечная точки отрезка имеют целочисленные координаты. Самый простой способ перевода отрезка в растровую форму заключается в пошаговом увеличении x , вычислении $y = mx + b$ и подсвечивании пикселей в точке $(x, \text{ROUND}(y))$. Вычисление произведения mx требует процессорного времени компьютера. Более того, для обеспечения достаточной точности придётся воспользоваться представлением данных с плавающей точкой. А при непосредственном выводе пикселей в растр будет осуществляться операция округления ROUND. Перечисленные недостатки такого алгоритма сводят на нет его эффективность в формировании насыщенных отрезками прямых растровых изображений.

Для повышения эффективности предложенного алгоритма можно исключить операцию умножения, если заметить, что при $dx = 1$ $m = dy/dx$ сводится к $m = dy$, т.е. изменение x на 1 приводит к изменению y на m (тангенс угла наклона отрезка).

Таким образом, если $x_{i+1} = x_i + 1$, то $y_{i+1} = y_i + m$ для всех точек (x_i, y_i) отрезка, т.е. последующие значения x и y определяются, исходя из предыдущих значений. Если $m > 1$, то единичный шаг по x будет приводить к шагу по y , большему 1. Поэтому следует x и y поменять ролями, придавая y единичное приращение, а x будет увеличиваться на $dx = dy/m = 1/m$. Подобный алгоритм растрового представления отрезка прямой назван *пошаговым алгоритмом*. Но и в этом случае мы не добиваемся ощутимого прироста в быстродействии алгоритма. Остаются вещественные значения оперируемых величин y и m и операция округления ROUND при выводе пикселей в растр.

Чтобы устранить вышеперечисленные недостатки, применяется целочисленный алгоритм Брезенхем [21] для построения отрезков прямых.

Пусть на бинарной сетке растра графического устройства вывода задан отрезок прямой (начальная точка (X_1, Y_1) и конечная точка (X_2, Y_2)), тангенс угла наклона которого находится в диапазоне $0 - 1$. Для построения отрезка используются значения управляющей переменной d_i . На каждом шаге построения отрезка прямой значение переменной d_i пропорционально разности значений расстояний s и t (см. рис. 1.1). Для i -го шага построения

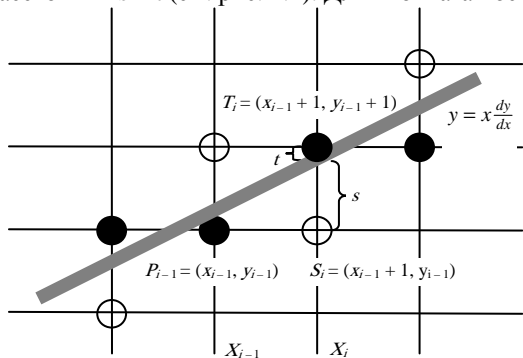


Рис. 1.1. Возможные точки для построения отрезка прямой

отрезка прямой, когда пиксель P_{i-1} уже известен, требуется определить какую из точек T_i или S_i необходимо выбрать. Если $s < t$ ($s - t < 0$), то выбираем точку S_i , как наиболее близко расположенную точку к реальному векторному изображению отрезка, в противном случае T_i .

На первом шаге работы алгоритма определяется начальное значение управляющей переменной как $d_1 = 2dy - dx$, где $dy = Y_2 - Y_1$ и $dx = X_2 - X_1$. Для каждого нового значения $x_i = x_{i-1} + 1$ проверяем знак управляющей переменной d_i :

- а) если $d_i > 0$, то выбирается пиксель T_i , тогда $y_i = y_{i-1} + 1$ и для следующего шага $d_{i+1} = d_i + 2(dy - dx)$
- б) если $d_i < 0$, то выбирается пиксель S_i , тогда $y_i = y_{i-1}$ и для следующего шага $d_{i+1} = d_i + 2dy$.

Например, для отрезка, проведенного из точки $(5, 8)$ в точку $(9, 11)$, последовательными значениями d_i будут $2, 0, -2, 4$ и 2 . Установленные пиксели и идеальное прохождение отрезка показаны на рис. 1.2.

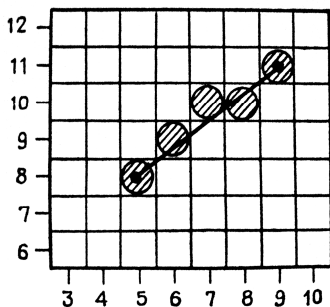


Рис. 1.2. Отрезок прямой

Для реализации алгоритма Брезенхем требуются только целые числа и минимальные арифметические целочисленные операции процессора ЭВМ: они включают сложение, вычитание и сдвиг влево (для осуществления операции умножения на два), что хорошо отражается на быстродействии построения отрезка прямой.

2. АЛГОРИТМ БРЕЗЕНХЕМА ДЛЯ РАСТРОВОЙ РАЗВЕРТКИ ОКРУЖНОСТИ

Существует несколько простых способов преобразования окружностей в растровую форму. Рассмотрим для простоты окружность с центром в начале координат, для которой $x^2 + y^2 = R^2$. Решая это уравнение относительно y , получим $y = \pm\sqrt{r^2 - x^2}$.

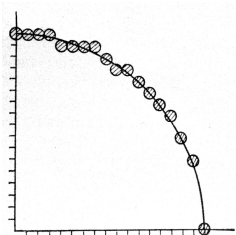


Рис. 2.1. Четвёртая часть окружности построенной по её уравнению

Чтобы изобразить четвертую часть окружности, можно увеличить x с единичным шагом от 0 до R и на каждом шаге вычислять положительные значения y (остальные четверти изображаются симметрично). Этот метод работоспособен, но не эффективен, поскольку в него входят операции умножения и извлечения квадратного корня. Более того, при значениях x , близких к R , в изображении окружности появляются заметные незаполненные промежутки (рис. 2.1).

Брезенхем разработал пошаговый алгоритм построения дуг [21], который более эффективен, чем метод построения окружности по его графику.

Схема алгоритма Брезенхема рассматривается на случай обхода лишь дуги окружности в 45° от $x = 0$ до $x = R/\sqrt{2}$. Для построения полной окружности используется восьмисторонняя симметрия (рис. 2.2).

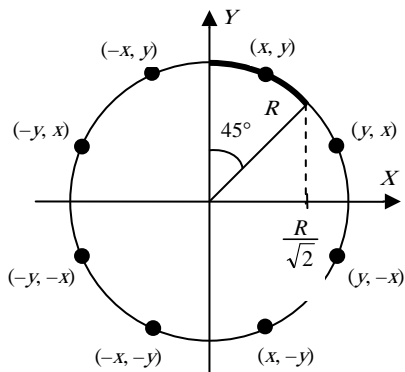


Рис. 2.2. Восьмисторонняя симметрия и область работы алгоритма

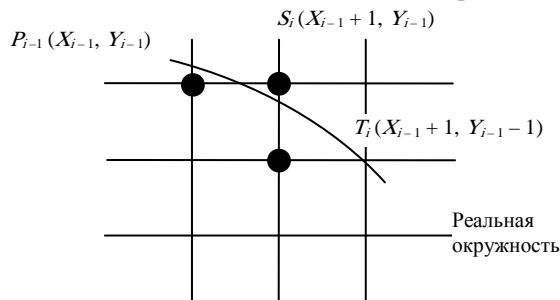


Рис. 2.3. Контрольные точки для окружности Брезенхема

Работа алгоритма аналогична построению отрезков прямых. Пиксели окружности (дуги) высвечиваются в зависимости от знака управляющей переменной d_i , где i – шаг работы алгоритма.

На каждом шаге построения окружности алгоритм Брезенхема выбирает точку $P(x_i, y_i)$, которая является наиболее ближайшей к истинной окружности, и поэтому ошибка $D(P_i) = (x_i^2 + y_i^2) - R^2$ будет близкой к нулю; т.е. значение $|D(P_i)|$ минимизируется на каждом шаге.

Пусть на $i - 1$ -ом шаге работы алгоритма точка P_{i-1} была выбрана как ближайшая к окружности при параметрах (x_{i-1}, y_{i-1}) . Далее, определяется какая из двух точек T_i или S_i расположена ближе к окружности $x_{i-1} + 1$:

- а) если $d_i < 0$, то выбирается пиксель $S_i(X_{i-1} + 1, Y_{i-1})$ и для следующего шага $d_{i+1} = d_i + 4x_{i-1} + 6$;
- б) если $d_i \geq 0$, то выбирается пиксель $T_i(X_{i-1} + 1, Y_{i-1} - 1)$ и для следующего шага $d_{i+1} = d_i + 4(x_{i-1} - y_{i-1}) + 10$.

На первом шаге работы алгоритма значение управляющей переменной d_1 определяется как $d_1 = 3 - 2R$.

Как и в алгоритме Брезенхема для построения отрезков прямых, в данном отсутствуют вещественные числа, операция округления.

3. АФФИННЫЕ ПРЕОБРАЗОВАНИЯ НАД ДВУМЕРНЫМИ ОБЪЕКТАМИ

В компьютерной графике часто применяются различные геометрические преобразования: вращение, сжатие, параллельный перенос и др. С помощью них, однократно описав топологию объекта, мы можем получить много различных его экземпляров. На практике подобные геометрические изменения (2D) объектов на плоскости описываются аффинными преобразованиями, описываемые следующими соотношениями [23]:

$$\begin{aligned} x' &= Ax + By + C; \\ y' &= Dx + Ey + F, \end{aligned}$$

где A, B, \dots, F – константы; x, y – координаты до преобразования; x', y' – новые координаты точек 2D объектов.

Рассмотрим основные 2D преобразования:

Перемещение (сдвиг) 2D объекта на D_x единиц по оси x и на D_y единиц по оси y (рис.

3.1) описывается следующими уравнениями:

$$\begin{cases} x' = x + D_x; \\ y' = y + D_y. \end{cases}$$

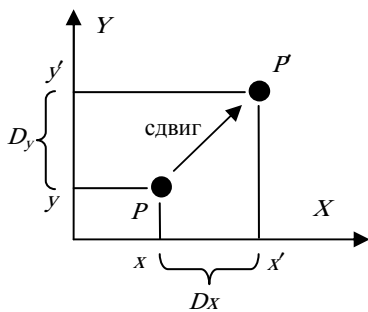


Рис. 3.1. Перемещение (сдвиг)

Используя однородные координаты, операция перемещения в векторной форме имеет вид

$$[x' \ y' \ 1] = [x \ y \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix}$$

или $P = PT(D_x, D_y)$, где $T(D_x, D_y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix}$.

Однородные координаты [23] позволяют все три преобразования (сдвиг, масштабирование и поворот) над $2D$ объектом реализовать с помощью единственной операции умножения вектора на матрицу, что повышает эффективность работы графического приложения.

В однородных координатах точка $P(x, y)$ записывается как $P(xW, yW, W)$ для любого масштабного множителя $W \neq 0$. При этом если для точки задано её представление в однородных координатах $P(X, Y, W)$, то можно найти её двумерные декартовы координаты как $x = X/W$ и $y = Y/W$. В наших работах W всегда равен 1, поэтому операция деления не требуется. Однородные координаты можно представить как вложение промасштабированной с коэффициентом W двумерной плоскости в плоскость $z = W$ (здесь $z = 1$) в трёхмерном пространстве.

Масштабирование $2D$ объекта в S_x раз по оси x и в S_y раз по оси y определяется уравнениями (рис. 3.2):

$$\begin{cases} x' = x S_x; \\ y' = y S_y. \end{cases}$$

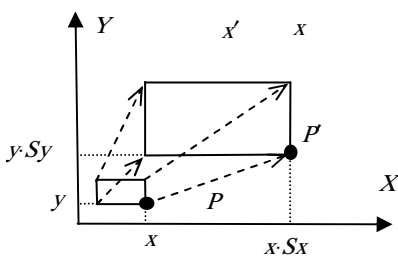


Рис. 3.2. Масштабирование

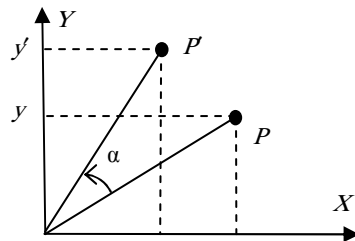


Рис. 3.3. Поворот

В однородных координатах матричная форма масштабирования имеет вид

$$[x' \ y' \ 1] = [x \ y \ 1] \cdot \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

или $P = P \cdot S(S_x, S_y)$, где $S(S_x, S_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

Поворот $2D$ объекта вокруг центра координат $(0, 0)$ на угол α (рис. 3.3) определяется следующим образом:

$$\begin{cases} x' = x \cos(\alpha) - y \sin(\alpha); \\ y' = x \sin(\alpha) + y \cos(\alpha). \end{cases}$$

В однородных координатах матричная форма поворота имеет вид

$$[x' \ y' \ 1] = [x \ y \ 1] \cdot \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

или $P = PR(\alpha)$, где $R(\alpha) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

Композиция двумерных преобразований выражается в объединении нескольких $2D$ преобразований.

Например, требуется повернуть $2D$ объект (рис. 3.4) на α градусов относительно произвольной точки $P_1(x_1, y_1)$. Поскольку известно лишь, как поворачивать двумерный объект вокруг начала координат, разобьём исходную задачу на три более простых:

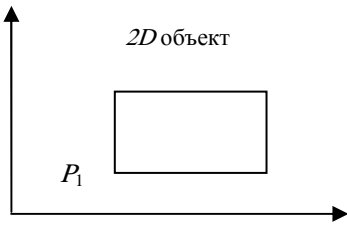


Рис. 3.4. Исходный 2D объект

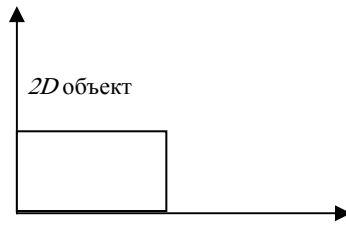


Рис. 3.5. После переноса в начало координат

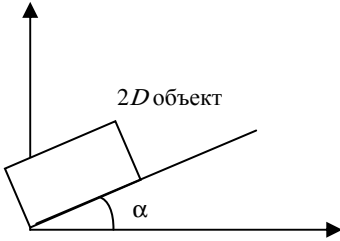


Рис. 3.6. После поворота на угол α

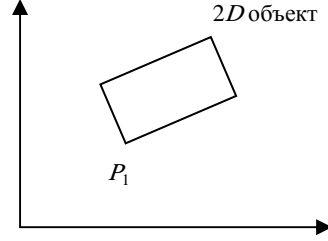


Рис. 3.7. Перенос 2D объекта в конечную позицию P_1

1. **Перенос** 2D объекта (рис. 3.5), при котором точка, находящаяся по координатам P_1 , совместится с началом координат:

$$\begin{cases} X = x - x_1; \\ Y = y - y_1. \end{cases}$$

2. **Поворот** на α градусов 2D объекта (рис. 3.6) относительно начала координат:

$$\begin{cases} X' = X \cos(\alpha) - Y \sin(\alpha); \\ Y' = X \sin(\alpha) + Y \cos(\alpha). \end{cases}$$

3. **Перенос** 2D объекта (рис. 3.7) в первоначальное положение точкой P_1 :

$$\begin{cases} x' = X' + x_1; \\ y' = Y' + y_1. \end{cases}$$

Если объединить математически проведённые преобразования, то получим результат:

$$\begin{cases} x' = (x - x_1) \cos(\alpha) - (y - y_1) \sin(\alpha) + x_1; \\ y' = (x - x_1) \sin(\alpha) + (y - y_1) \cos(\alpha) + y_1. \end{cases}$$

В матричной форме данные операции имеют вид

$$\begin{aligned} \begin{bmatrix} x' & y' & 1 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_1 & -y_1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_1 & y_1 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ x_1 \cdot (1 - \cos(\alpha)) + y_1 \cdot \sin(\alpha) & y_1 \cdot (1 - \cos(\alpha)) - x_1 \cdot \sin(\alpha) & 1 \end{bmatrix}. \end{aligned}$$

Преобразование координат объекта после подобной композиции в символьных выражениях имеет вид:

$$P' = P \cdot T(-x_1, -y_1) \cdot R(\alpha) \cdot T(x_1, y_1).$$

4. ТРЁХМЕРНЫЕ АФФИННЫЕ ПРЕОБРАЗОВАНИЯ 3D ОБЪЕКТОВ

Аффинные преобразования 3D объектов имеют вид [23]:

$$\begin{cases} x' = Ax + By + Cz + D; \\ y' = Ex + Fy + Gz + H; \\ z' = Kx + Ly + Mz + N, \end{cases}$$

где A, B, \dots, N – константы; x, y, z – координаты до преобразования; x', y', z' – новые координаты точек 3D объектов.

Рассмотрим основные трёхмерные преобразования 3D объекта относительно начала системы координат:

1. **Перемещение** (сдвиг) на Dx, Dy, Dz записывается

$$\begin{cases} x' = x + Dx; \\ y' = y + Dy; \\ z' = z + Dz. \end{cases}$$

2. Масштабирование в S_x, S_y, S_z раз имеет вид

$$\begin{cases} x' = S_x x; \\ y' = S_y y; \\ z' = S_z z. \end{cases}$$

3. Поворот вокруг оси Z на угол α (рис. 4.1) описывается выражениями

$$\begin{cases} x' = x \cos(\alpha) - z \sin(\alpha); \\ y' = x \sin(\alpha) + y \cos(\alpha); \\ z' = z. \end{cases}$$

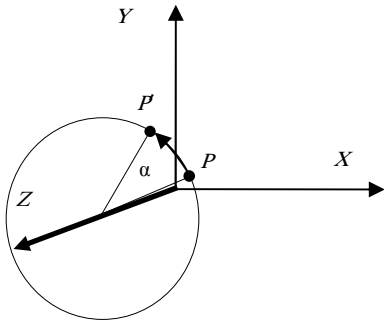


Рис. 4.1. Поворот вокруг оси Z на угол α

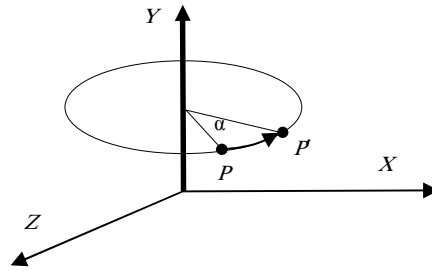


Рис. 4.2. Поворот вокруг оси Y на угол α

4. Поворот вокруг оси Y на угол α (рис. 4.2) описывается выражениями

$$\begin{cases} x' = x \cos(\alpha) - z \sin(\alpha); \\ y' = y; \\ z' = x \sin(\alpha) + z \cos(\alpha). \end{cases}$$

5. Поворот вокруг оси X на угол α (рис. 4.3) описывается выражениями

$$\begin{cases} x' = x; \\ y' = y \cos(\alpha) - z \sin(\alpha); \\ z' = y \sin(\alpha) + z \cos(\alpha). \end{cases}$$

Подобно тому, как двумерные преобразования описываются матрицами размером 3×3 , трёхмерные преобразования могут быть представлены матрицами размером 4×4 . В этом случае трёхмерная точка (x, y, z) записывается

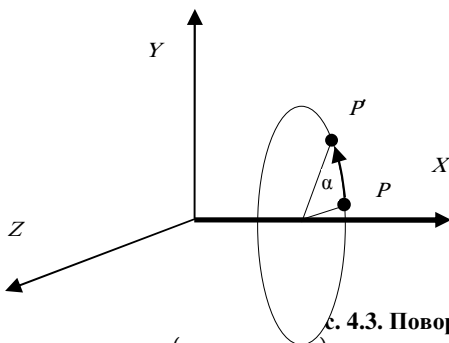


Рис. 4.3. Поворот вокруг оси X на угол α

в однородных координатах как (wx, wy, wz, w) , где $w \neq 0$. Для получения декартовых координат из однородных необходимо первые три параметра точки разделить на w . На практике $w = 1$. В однородных координатах матрицы основных преобразований бывают

$$T(Dx, Dy, Dz) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Dx & Dy & Dz & 1 \end{bmatrix}.$$

1. Перемещения –

в символьной форме данное преобразование записывается

$$P' = PT(Dx, Dy, Dz),$$

где P' – новое значение координат; P – старое значение координат 3D объекта;

$$S(Sx, Sy, Sz) = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

2. Масштабирования –

в символьной форме данное преобразование – масштабирование имеет вид:

$$P' = PS(Sx, Sy, Sz),$$

где P' – новое значение координат; P – старое значение координат 3D объекта.

3. Поворота относительно оси X:

$$Rx(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) & 0 \\ 0 & -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

4. Поворота относительно оси Z:

$$Rz(\alpha) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

5. Поворота относительно оси Y:

$$Ry(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

5. ПРОЕКЦИРОВАНИЕ ТРЁХМЕРНЫХ ОБЪЕКТОВ

Для того чтобы увидеть на плоскости монитора трёхмерное изображение, используют эффект проецирования. В общем случае проекции преобразуют точки, заданные в системе координат размерностью n , в точки системы координат размерностью меньшей, чем n . В рассматриваемом случае точки трёхмерного пространства, заданные в мировой системе координат, преобразуются в точки двумерного пространства – экранных координат.

Проекция строится с помощью проецирующих лучей или проекторов, которые выходят из точки, называемой центром проекции. Проекторы проходят через плоскость проекционную или картинную плоскость и затем проходят через каждую точку трёхмерного объекта и образуют тем самым проекцию.

Геометрические проекции делятся на два вида: центральные и параллельные. Если центр проекции находится на конечном расстоянии от проекционной плоскости, то проекция – центральная. Если же центр проекции находится в бесконечности, то проекция – параллельная (рис. 5.1).

Точкой схода называется точка пересечения центральных проекций любой совокупности параллельных прямых, которые не параллельны проекционной плоскости. Существует бесконечное множество точек схода. Точка схода называется главной, если совокупность прямых параллельна одной из координатных осей. В зависимости от того, сколько координатных осей пересекает проекционную плоскость различают одно-, двух- и трёхточечные проекции (рис. 5.2).

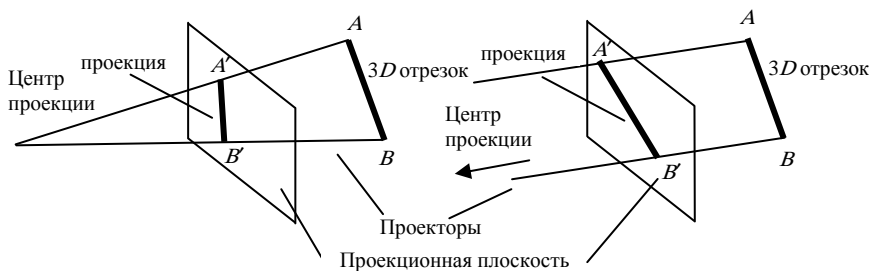


Рис. 5.1. Центральная и параллельная проекции

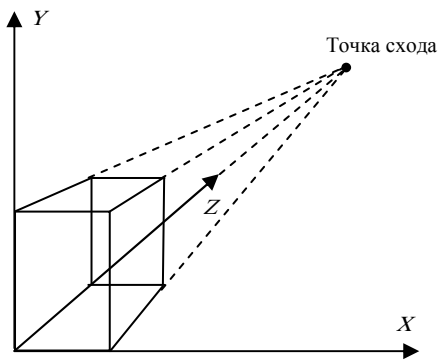


Рис. 5.2. Одноточечная проекция

Простейшей является параллельная прямоугольная проекция. В ней совместно изображаются виды сверху, спереди и сбоку. Эти проекции часто используются в черчении. В зависимости от соотношения между направлениями проецирования и нормалью к проекционной плоскости параллельные проекции разделяются на ортографические или ортогональные, в которых эти направления совпадают, и косоугольные, в которых они не совпадают. В зависимости от положения осей системы координат объекта относительно проекционной плоскости ортографические проекции делятся на аксонометрические и изометрические. В изометрических проекциях оси системы координат составляют одинаковые углы с проекционной плоскостью. В аксонометрических проекциях эти углы разные. Центральная проекция приводит к визуальному эффекту, подобному тому, который даёт зрительная система человека. При этом наблюдается эффект перспективного укорачивания, когда размер проекции объекта изменяется обратно пропорционально расстоянию от центра проекции до объекта. В параллельных проекциях отсутствует перспективное укорачивание, за счёт чего изображение получается менее реалистичным и параллельные прямые всегда остаются параллельными.

В компьютерной графике различают две модели центральной проекции:

- проекционная плоскость перпендикулярна оси Z и совпадает с плоскостью $Z = d$ (рис. 5.3, вариант 1);
- проекционная плоскость перпендикулярна оси Z и совпадает с плоскостью $Z = 0$ (рис. 5.3, вариант 2).

В первом случае $3D$ точка P проецируется на экран как P' . Расстояние от наблюдателя до проекционной плоскости равно d . Необходимо определить координаты точки P' на экране. Обозначим их x' и y' . Из подобия треугольников $P_y P_z O$ и $y_p d O$ находим, что

$$\frac{y}{z} = \frac{y_p}{d}, \Rightarrow y_p = \frac{yd}{z}. \quad (5.1)$$

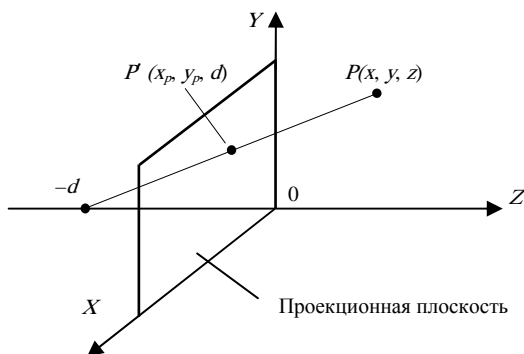
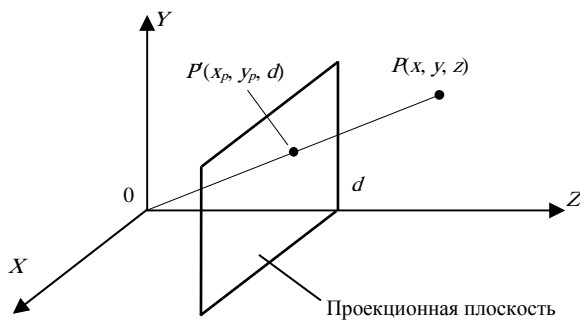


Рис. 5.3. Центральная проекция: варианты 1 и 2

Аналогично для x : $x_p = \frac{xd}{z}$.

Напомним, что d – это расстояние наблюдателя до проекционной плоскости, а наблюдатель находится в точке $(0, 0, 0)$.

Если поместить проекционную плоскость в начало координат, а точку наблюдения на расстояние $-d$, как показано на рис. 5.3 (вариант 2), то формулы для x_p и y_p примут вид:

$$x_p = \frac{x}{1+(z/d)}; \quad y_p = \frac{y}{1+(z/d)}. \quad (5.2)$$

Формулы (5.2) более удобны при необходимости простым образом приближать или удалять наблюдателя от проекционной плоскости. Формулы (5.1) требуют меньше времени для вычислений за счёт отсутствия операции сложения.

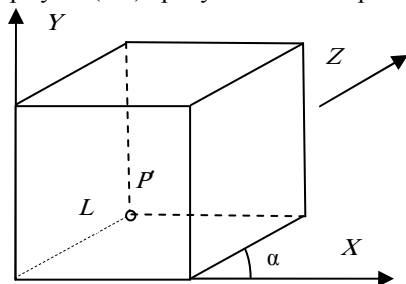


Рис. 5.4. Косоугольная параллельная проекция единичного куба

Для косоугольных проекций (рис. 5.4) проекторы пересекают проекционную плоскость под непрямым углом. Для военной косоугольной проекции этот угол составляет 45 градусов, а для кабинетной – 63,4 градуса. Математические соотношения формул косоугольной проекции произвольного 3D объекта базируются на формулах косоугольной проекции единичного куба (ребро длиной 1).

Из рисунка 5.4 видно, что проекцией точки $P(0, 0, 1)$, находящейся на задней стороне единичного куба, является точка $P'(L \cos(\alpha), L \sin(\alpha))$, принадлежащая, плоскости XY .

Математические соотношения для косоугольной параллельной проекции произвольного 3D объекта имеют вид

$$x_p = x + z(L \cos(\alpha)); \quad y_p = y + z(L \sin(\alpha)).$$

Для военной проекции $L = 1$, а для кабинетной проекции $L = 0,5$. Угол α обычно выбирают 30° или 45° . Не следует путать этот угол с углом между проекторами и проекционной плоскостью.

6. РАСТРОВОЕ ТЕКСТУРИРОВАНИЕ

Для отображения трёхмерных объектов часто используются полигональные поверхности, каждая грань отображается с наложенной текстурой [23]. Задача текстурирования формулируется таким образом: есть грань – согласно предположениям, треугольная. И имеется изображение текстуры в виде двумерного массива значений цветов точек. Для каждой точки полигона (x, y) требуется найти её цвет из соответствующей точки текстуры. А для этого необходимо найти координаты текстуры (x_t, y_t) для этой точки.

Преобразование координат точек из значений (x, y) в (x_t, y_t) может быть, например, аффинное:

$$\begin{aligned} x_t &= Ax + By + C; \\ y_t &= Dx + Ey + F, \end{aligned}$$

где коэффициенты A, B – константы при пересчёте координат всех пикселей для отдельной текстурированной грани.

Такое преобразование координат можно использовать, если привязать текстуру к грани по трём опорным точкам (см. рис. 6.1).

Связь текстуры с гранью по трём точкам соответствует уравнениям:

$$\begin{aligned} x_{ti} &= Ax_i + By_i + C; \\ y_{ti} &= Dx_i + Ey_i + F, \end{aligned}$$

где $i = 1, 2, 3$.

По известным координатам (x_{ti}, y_{ti}) и (x_i, y_i) можно найти коэффициенты A, B, \dots, F , если решить систему линейных уравнений. Эта система распадается на две независимые системы третьего порядка. Для упрощения записи заменим x_{ti} на X_i , а y_{ti} на Y_i :

$$\begin{aligned} X_1 &= Ax_1 + By_1 + C, & Y_1 &= Dx_1 + Ey_1 + F, \\ X_2 &= Ax_2 + By_2 + C, & Y_2 &= Dx_2 + Ey_2 + F, \\ X_3 &= Ax_3 + By_3 + C, & Y_3 &= Dx_3 + Ey_3 + F. \end{aligned}$$

Для решения систем линейных уравнений известно множество способов. Используем способ, основанный на вычислении определителей. Запишем решение системы в следующем виде:

$$\begin{aligned} A &= \det A / \det = (X_1(y_2 - y_3) + X_2(y_3 - y_1) + X_3(y_1 - y_2)) / \det; \\ B &= \det B / \det = (x_1(X_2 - X_3) + x_2(X_3 - X_1) + x_3(X_1 - X_2)) / \det; \\ C &= X_1 - Ax_1 - By_1. \end{aligned}$$

Таким же способом решаем систему уравнений для коэффициентов D, E, F .

$$D = \det D / \det = (Y_1(y_2 - y_3) + Y_2(y_3 - y_1) + Y_3(y_1 - y_2)) / \det;$$

$$E = \det E / \det = (x_1(Y_2 - Y_3) + x_2(Y_3 - Y_1) + x_3(Y_1 - Y_2)) / \det;$$

$$F = Y_1 - Dx_1 - Ey_1.$$

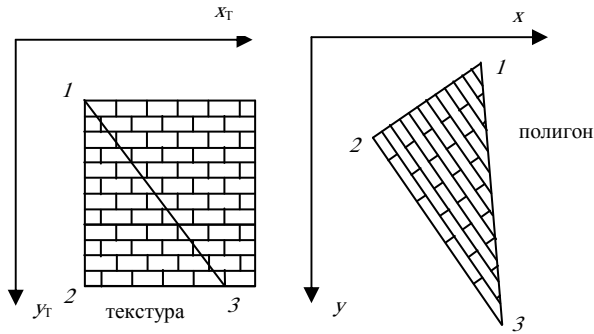


Рис. 6.1. Наложение текстуры на грань объекта

Вышеприведённые математические соотношения справедливы только для аксонометрических проекций 3D объектов. Для текстурирования перспективных проекций необходимы дополнительные математические операции.

В перспективных проекциях преобразования координат из видовых координат в координаты плоскости проецирования нелинейны. Поэтому экранные координаты вначале следует преобразовать в такие, которые линейно связаны с мировыми – это могут быть, скажем, видовые. А затем видовые координаты (или непосредственно мировые) связать с координатами текстуры аффинным преобразованием, используя, например, метод трех точек [23].

7. АЛГОРИТМ «ПЛАВАЮЩЕГО ГОРИЗОНТА» ДЛЯ ВИЗУАЛИЗАЦИИ ФУНКЦИОНАЛЬНО-ЗАДАННЫХ ПОВЕРХНОСТЕЙ

Алгоритм «плавающего горизонта» [22] используется для удаления невидимых линий трёхмерного представления функции, описывающей поверхность в виде $F(x, y, z) = 0$.

В данном алгоритме трёхмерная задача сводится к двумерной путём пересечения исходной поверхности последовательностью параллельных секущих плоскостей, имеющих постоянные значения координат x , y или z .

Поверхность складывается из последовательности кривых, лежащих в каждой из этих плоскостей. Предполагается, что полученные кривые являются однозначными функциями независимых переменных (рис. 7.1).

Обычно используется функция вида $y = f(x, z)$, где z – постоянно на каждой из заданных параллельных плоскостей.

Если спроецировать полученные кривые на плоскость $z = 0$ (рис. 7.2), то становится понятна идея алгоритма удаления невидимых участков заданной поверхности.

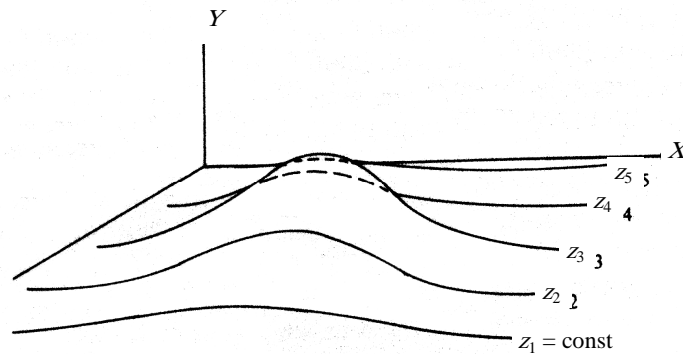


Рис. 7.1. Кривые в секущих плоскостях

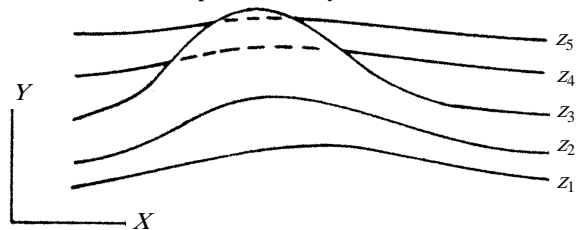


Рис. 7.2. Проекция кривых на плоскость $Z = 0$

Для каждой пересекающей плоскости, начиная с ближайшей к точке наблюдения, строится кривая, лежащая на ней, т.е. для каждого значения координаты x в пространстве изображения определяется соответствующее значение y .

Удаление невидимых линий заключается в следующем: если на текущей плоскости при некотором заданном значении x соответствующее значение y на кривой больше значения y для всех предыдущих кривых при этом значении x , то текущая кривая видима в этой точке; в противном случае она невидима.

Реализация данного алгоритма достаточно проста. Для хранения максимальных значений y при каждом значении x используется массив, длина которого равна числу различных точек (разрешению) по оси x в пространстве изображения. Значения, хранящиеся в этом массиве, представляют собой текущие значения «горизонта». Поэтому по мере рисования каждой оче-

редной кривой этот горизонт «всплывает». Фактически этот алгоритм удаления невидимых линий работает каждый раз с одной линией.

Алгоритм работает очень хорошо до тех пор, пока какая-нибудь очередная кривая не окажется ниже самой первой из кривых.

Подобные кривые, естественно, видимы и представляют собой нижнюю сторону исходной поверхности, однако алгоритм будет считать их невидимыми. Нижняя сторона поверхности делается видимой, если модифицировать этот алгоритм, включив в него нижний горизонт, который опускается вниз по ходу работы алгоритма. Это реализуется при помощи второго массива, длина которого равна числу различимых точек по оси x в пространстве изображения. Этот массив содержит наименьшие значения y для каждого значения x .

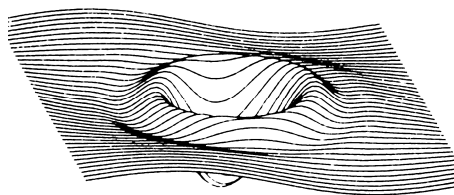


Рис. 7.3. Пример работы алгоритма «плавающего горизонта»

Алгоритм теперь становится таким: если на текущей плоскости при некотором заданном значении x соответствующее значение y на кривой больше максимума или меньше минимума по y для всех предыдущих кривых при этом x , то текущая кривая видима. В противном случае она невидима.

Пример поверхности функции:

$$y = (1/5) \sin(x) \cos(z) - (3/2) \cos(7a/4) \times e^{-a},$$

где $a = (x - \pi)^2 + (z - \pi)^2$, изображённой в интервале $(0, 2\pi)$, представлена на рис. 7.3.

8. АЛГОРИТМ УДАЛЕНИЯ НЕВИДИМЫХ ГРАНЕЙ, ИСПОЛЬЗУЮЩИЙ BSP-ДЕРЕВА

В процессе проектирования пространственных ($3D$) объектов, например, изделий машиностроения, архитектурных сооружений и т.п., поверхности которых представлены с помощью плоских многоугольников (полигонов), приходится решать множество задач компьютерной графики, одной из которых является определение видимости полигонов сцены $3D$ объектов камерой наблюдателя.

В случае обработки неподвижной визуализационной сцены задача удаления невидимых полигонов не является предметом детального изучения специалистов в области компьютерной графики из-за того, что нет временных ограничений в реализации данного процесса. Но как только требуется наблюдать за сценой в динамике, то задаче удаления невидимых полигонов (или их фрагментов) уделяется первостепенное внимание.

Одним из эффективных алгоритмов удаления невидимых граней в режиме реального времени является алгоритм, использующий BSP-дерево [26]. Под BSP-деревом понимается пространственное взаимоотношение полигонов $3D$ сцены.

Концептуально BSP-дерево формируется следующим образом. Пусть какая-то плоскость в объектном пространстве разбивает множество полигонов сцены на два подмножества, и при этом ни один из полигонов не попал под рассеечение плоскостью. Тогда очевидно, что подмножество полигонов, не содержащее наблюдателя, не может экранировать полигоны из множества с наблюдателем. Если в каждом подмножестве полигонов опять найдем плоскости, не пересекающие внутренние полигоны, то данные подмножества будут разбиты на две части. Продолжая процесс разбиения подмножеств полигонов, придём к размеру подмножества в один полигон. Процесс разбиения сцены на подмножества с параллельным формированием BSP-дерева осуществляется рекурсивно.

Совокупность плоскостей и соответствующих подмножеств дают в результате двоичное дерево разбиения пространства. Узлами данного дерева будут разбивающие плоскости, а ветви – соответствующие подмножества. Листьями BSP-дерева будут полигоны объекта визуализации.

На практике в качестве разбивающей плоскости выбирают плоскость, проходящую через один из полигонов. Плоскость может рассечь несколько полигонов на части, после чего рассеченные полигоны подменяются их составными частями в модели данных объекта. И полученные таким образом новые полигоны распределяются в соответствующие подмножества BSP-дерева.

Для определения расположения полигонов по отношению к разбивающей плоскости можно воспользоваться значением тестовой функции – $TF(x, y, z)$, которая имеет вид

$$TF = Ax + By + Cz + D,$$

где A, B, C, D – коэффициенты уравнения разбивающей плоскости; x, y, z – координаты анализируемой точки в пространстве сцены.

Если $TF \geq 0$, то это указывает на правое подмножество BSP-дерева, а $TF < 0$ указывает на левое подмножество дерева.

Процесс удаления невидимых полигонов сцены с использованием BSP-дерева для построения итогового изображения заключается в рекурсивном проходе дерева от «передних» ветвей к «задним» по следующей схеме:

- начиная с корневого узла дерева, определяем положение наблюдаемой камеры относительно рассекающей плоскости полигона данного узла;
- прорисовываем противоположную ветвь, потом полигоны, принадлежащие рассекающей плоскости и, в последнюю очередь, ветвь, в которой находится наблюдатель;
- рекурсивно повторяем этот процесс для каждого узла BSP-дерева сцены визуализации.

При построении BSP-дерева в качестве корневого узла может быть выбран любой полигон сцены, но исследование показало, что в зависимости от того, какой полигон выбран, вид дерева меняется, что в конечном итоге сказывается на процессе обработки дерева при построении изображения.

В процессе работ построения BSP-деревьев 3D сцен выявилось, что стремление получить сбалансированное дерево сцены визуализации, как правило, приводит к разрастанию модели данных сцены за счёт деления полигонов на составные части секущими плоскостями узловых полигонов дерева.

Для получения некоторого компромисса между сбалансированностью BSP-дерева сцены и небольшим количеством возможных рассечений полигонов в работе предлагается следующая методика.

Сформируем таблицу состояний для каждого полигона анализируемой 3D сцены со следующими параметрами:

- количество полигонов сцены, находящихся левее данного [«←»];
- количество полигонов сцены, находящихся правее данного [«→»];
- количество пересечений полигонов сцены с плоскостью данного полигона [«Кол. пересечений»];
- значение разности по модулю первого и второго параметра [$\text{abs}(\text{«←»} - \text{«→»})$].

Последний параметр таблицы указывает на сбалансированность дерева. При минимальном значении этого поля разность между количеством полигонов правой части дерева и левой минимальна.

Понятие «левее» и «правее» для расположения полигонов чисто условное, так как на самом деле здесь играет роль знака пробной функции секущей плоскости, куда подставляем координаты вершин анализируемых полигонов.

Для каждого подмножества полигонов сцены строим подобные таблицы, из которых выбираем полигон с минимальным значением поля – [$\text{abs}(\text{«←»} - \text{«→»})$] и минимальным значением поля – [«Кол. пересечений»]. Подобные полигоны будут являться узловыми полигонами строящегося BSP-дерева анализируемой сцены. Процесс разбиения полигонов на подмножества завершается, когда в таблице состояний находятся всего два полигона.

Рассмотрим некоторый пример 3D сцены, состоящей из четырёх полигонов: 1, 2, 3 и 4 (рис. 8.1).

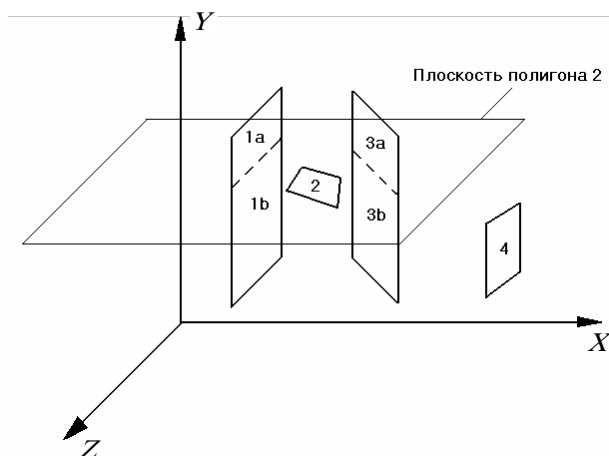


Рис. 8.1. Пример пространственной сцены

Построим BSP-дерево данной сцены. Для организации верхнего уровня BSP-дерева заполним таблицу состояний полигонов в сцене (табл. 8.1).

Таблица 8.1

№ полигона	«←»	«→»	Кол. пересечений	$\text{abs}(\text{«←»} - \text{«→»})$
1	0	3	0	3
2	2	3	2	1
3	2	1	0	1
4	3	0	0	3

В этой таблице выделяем полигоны с минимальным значением поля – [$\text{abs}(\text{«←»} - \text{«→»})$] и выбираем из них тот, для которого количество пересечений минимальное. В данном примере наиболее близко для корневого сегмента BSP-дерева подходят полигоны 2 и 3. С точки зрения балансировки дерева вариант с корневым полигоном 2 наиболее предпочтителен, но в этом случае приходится разбивать два полигона (1 и 3) на части, поэтому выбираем полигон 3, так как количество пересечений равно нулю.

При появлении в таблице полигонов с одинаковыми значениями четвертого параметра – [$\text{abs}(\text{«←»} - \text{«→»})$] выбираем из этого списка полигон, для которого значение поля – [«Кол. пересечений»] минимальное.

Подобные таблицы строятся для каждого левого и правого подмножества полигонов относительно выбранного полигона строящегося дерева. И по аналогичной схеме определяется внутренний узловой полигон. В данном случае в левом подмножестве останется полигон 1 и 2, а в правом 4. Для полигонов 1 и 2 опять строим таблицу состояний (табл. 8.2).

Таблица 8.2

№ полигона	«-»	«+»	Кол. пересечений	Abs(«-» - «+»)
1	0	1	0	1
2	1	0	1	1

Исходя из значений поля – [«Кол. пересечений»], выбираем полигон 1.

На этом процесс разбиения подмножеств полигонов 3D сцены заканчивается.

Параллельно с построением таблиц состояний подмножеств полигонов строится BSP-дерево сцены. Для рассматриваемой сцены возможные структуры BSP-дерева представлены на рис. 8.2.

В данном примере приходим к варианту дерева 3.

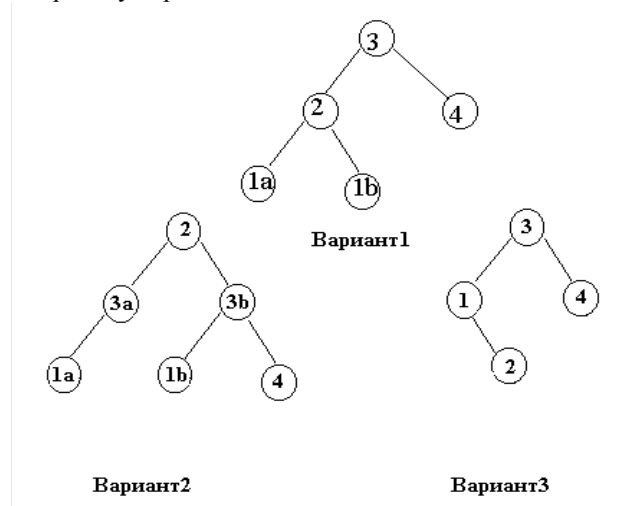


Рис 8.2. Возможные варианты BSP-дерева

Большой заслугой BSP-дерева в алгоритме удаления невидимых полигонов сцены заключается в том, что все построения бинарного дерева осуществляются заранее на этапе, не связанном с визуализацией. А в процессе визуализации сцены BSP-дерево используется как путеводитель в множестве полигонов сцены. Перемещение по готовым указателям для достижения соответствующего полигона – процесс быстрый и не требует сложных математических расчётов. Изменение положения камеры наблюдателя никоим образом не влияет на структуру BSP-дерева сцены, что для подсистем визуализации реального времени имеет главенствующее значение. Нет необходимости для каждого кадра визуализации сцены в динамике заниматься сортировкой полигонов, так как в BSP-дерева всё уже предусмотрено.

9. ПОСТРОЕНИЕ КРИВЫХ

Задачи построения кривых по точкам возникают в компьютерной графике при проектировании, обработке изображений и распознавании образов. Отыскание кривой, проходящей через заданное множество точек, составляет задачу интерполирования, а отыскание кривой, проходящей вблизи заданного множества точек, – задачу аппроксимации. Для точной передачи кривизны по исходным данным удобнее использовать интерполяционные кривые, а если требуется, заодно, и убрать «шум» в исходных данных, то – аппроксимирующие кривые.

Рассмотрим несколько кривых Безье. Пусть известны три точки: b_0 , b_1 и b_2 (рис. 9.1).

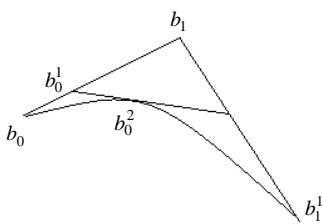


Рис. 9.1. Кривая по трём точкам

b_i^j строятся в зависимости от заданного отношения (на рисунке $t \sim 1/3$):

$$b_0^1(t) = (1-t)b_0 + tb_1;$$

$$b_1^1(t) = (1-t)b_1 + tb_2;$$

$$b_0^2(t) = (1-t)b_0^1(t) + tb_1^1(t), \text{ или}$$

$$b_0^2(t) = (1-t)^2 b_0 + 2t(1-t)b_1 + t^2 b_2.$$

$b^2(t)$ – парабола, вписанная в угол, т.е. при $0 \leq t \leq 1$. Кривая $b^2(t)$ всегда располагается внутри треугольника: $b_0 b_1 b_2$, $b^2(0) = b_0$, $b^2(1) = b_2$.

Рассмотрим ещё один случай – четырёх точек (рис. 9.2). Пусть заданы четыре точки: b_0 , b_1 , b_2 и b_3 .

В зависимости от параметра t (на рисунке $t \sim 1/3$), выбираем точки $b_0^1 b_1^1$ и b_2^1 . В аналогичном соотношении делим $b_0^1 b_1^1$ и $b_1^1 b_2^1$ и вписываем кривую b^3 .

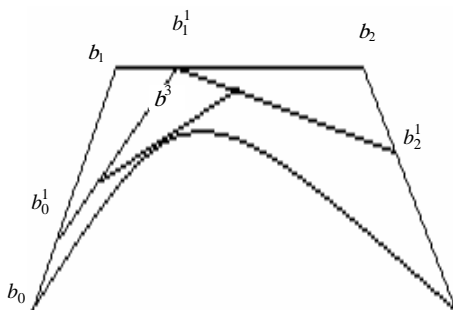


Рис. 9.2. Кривая по четырём точкам

Рассмотрим более обобщённый случай. Пусть заданы точки: b_0, b_1, \dots, b_m из E^3 и t – действительный параметр. Тогда кривая Безье b^p имеет вид [28]

$$b(t) = \sum_{i=0}^m C_m^i t^i (1-t)^{m-i} b_i,$$

где $0 \leq t \leq 1$; $C_m^i = \frac{m!}{i!(m-i)!}$ – число сочетаний из m по i .

Необходимо заметить, что данная кривая только в начальной (b_0) и конечной b_m опорной точке интерполирует исходные данные. Промежуточные точки кривая Безье аппроксимирует.

Широкое применение в компьютерной графике нашли себя параметрические кубические кривые. Такие кривые могут отобразить любые исходные данные. Например, параметрическими кубическими кривыми очень просто можно описать замкнутые или самопересекающиеся кривые или, например, когда у разных точек набора совпадают абсциссы.

Пусть t – независимый параметр, такой что $0 \leq t \leq 1$. Кубическим параметрическим сплайном назовём следующую систему уравнений:

$$\begin{cases} x(t) = a_x t^3 + b_x t^2 + c_x t + d_x; \\ y(t) = a_y t^3 + b_y t^2 + c_y t + d_y; \\ z(t) = a_z t^3 + b_z t^2 + c_z t + d_z. \end{cases}$$

Координаты точек на кривой описываются вектором $(x(t), y(t), z(t))$, а три производные задают координаты соответствующего касательного вектора в точке. Например, для координаты x :

$$\frac{dx}{dt} = 3a_x t^2 + 2b_x t + c_x.$$

Одним из способов задания параметрического кубического сплайна является указание координат начальной и конечной точек, а также векторов касательных в них. Такой способ задания называется формой Эрмита [21]. Обозначим концевые точки P_1 и P_4 , а касательные векторы в них R_1 и R_4 . Индексы выбраны таким образом с учётом дальнейшего изложения.

Будем решать задачу нахождения четвёрки коэффициентов a_x, b_x, c_x, d_x , так как для оставшихся двух уравнений коэффициенты находятся аналогично. Запишем условие для построения сплайна:

$$x(0) = P_{1x}, \quad x(1) = P_{4x}, \quad x'(0) = R_{1x}, \quad x'(1) = R_{4x}. \tag{9.1}$$

Перепишем выражение для x в векторном виде:

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}_x.$$

Обозначим вектор строку $T = [t^3, t^2, t, 1]$ и вектор столбец коэффициентов

$$C_x = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}_x, \quad \text{тогда } x(t) = TC_x.$$

Из (9.1) следует, что $x(0) = P_{1x} = [0, 0, 0, 1] C_x$, $x(1) = P_{4x} = [1, 1, 1, 1] C_x$.

Для касательных $x'(t) = [3t^2, 2t, 1, 0] C_x$, $\Rightarrow x'(0) = R_{1x} = [0, 0, 1, 0] C_x$, $x'(1) = R_{4x} = [3, 2, 1, 0] C_x$.

Отсюда получаем векторно-матричное уравнение

$$\begin{bmatrix} P_{1x} \\ P_{4x} \\ R_{1x} \\ R_{4x} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} C_x.$$

Эта система решается относительно C_x нахождением обратной матрицы размером 4×4 :

$$C_x = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_{1x} \\ P_{4x} \\ R_{1x} \\ R_{4x} \end{bmatrix} = M_h G_{hx}.$$

Здесь M_h – эрмитова матрица; G_h – геометрический вектор Эрмита. Подставим выражение C_x для нахождения $x(t)$: $x(t) = TM_h G_{hx}$.

Аналогично для остальных координат: $y(t) = TM_h G_{hy}$, $z(t) = TM_h G_{hz}$.

Выпишем в явном виде формулы для вычисления координат точек сплайна. Так как $TM_h = \left[(2t^3 - 3t^2 + 1), (-2t^3 + 3t^2), (t^3 - 2t^2 + t), (t^3 - t^2) \right]$, то умножая справа на G_{hx} , получаем

$$x(t) = TM_h G_{hx} = P_{1x}(2t^3 - 3t^2 + 1) + P_{4x}(-2t^3 + 3t^2) + R_{1x}(t^3 - 2t^2 + t) + R_{4x}(t^3 - t^2).$$

Форму кривой, заданной в форме Эрмита, легко изменять, если учитывать, что направление вектора касательной задаёт начальное направление, а модуль вектора касательной задаёт степень вытянутости кривой в направлении этого вектора, как показано на рис. 9.3.

Рассмотрим форму Безье, которая отличается от формы Эрмита способом задания граничных условий, а именно, вместо векторов R_1 и R_4 вводятся точки (и соответствующие им радиус векторы) P_2 и P_3 , как показано на рис. 9.4, такие, что выполняются условия:

$$P'(0) = R_1 = 3(P_2 - P_1) \quad \text{и} \quad P'(1) = R_4 = 3(P_4 - P_3).$$

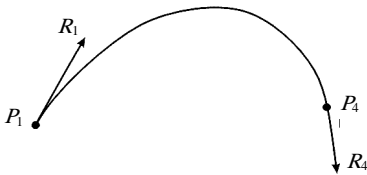


Рис. 9.3. Параметрический сплайн в форме Эрмита

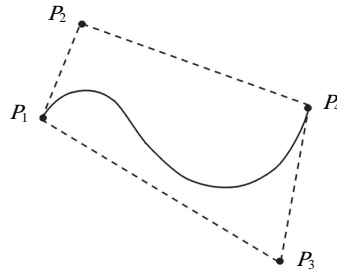


Рис. 9.4. Параметрический сплайн в форме Безье

Переход от формы Эрмита к форме Безье осуществляется преобразованием

$$G_h = \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = M_{hb} G_b,$$

где G_b – геометрический вектор Безье.

Подставляя это в выражение для $x(t)$, получаем

$$x(t) = TM_h G_{hx} = TM_h M_{hb} G_{bx} = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4.$$

Заметим, что матрица вида $M_h M_{hb} = M_b = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ – называется матрицей Безье [27].

10. ПОСТРОЕНИЕ ПАРАМЕТРИЧЕСКИХ КУБИЧЕСКИХ ПОВЕРХНОСТЕЙ

Существуют три основных способа представления поверхностей:

- 1) с помощью функций;
- 2) в параметрическом виде;
- 3) в полигональном виде.

Из класса параметрических поверхностей рассмотрим бикубические поверхности в форме Эрмита, Безье и В – сплайнов [21].

Бикубические поверхности задаются кубическими уравнениями от двух переменных s и t . Изменяя оба параметра от 0 до 1, можно определить все точки на куске поверхности. Если одному из параметров присвоить постоянное значение, а другой – изменять в диапазоне от 0 до 1, то в результате получим кубическую кривую. Для удобства мы будем рассматривать только уравнение для x :

$$x(s, t) = a_{11}s^3t^3 + a_{12}s^3t^2 + a_{13}s^3t + a_{14}s^3 + a_{21}s^2t^3 + a_{22}s^2t^2 + a_{23}s^2t + a_{24}s^2 + a_{31}st^3 + a_{32}st^2 + a_{33}st + a_{34}s + a_{41}t^3 + a_{42}t^2 + a_{43}t + a_{44}.$$

Запишем в более удобной форме

$$x(s, t) = SC_xT^T,$$

где $S = [s^3 \ s^2 \ s \ 1]$; $T = [t^3 \ t^2 \ t \ 1]$, а T^T – транспонированная матрица T .

Подобная запись называется алгебраической формой представления, так как C_x задаёт коэффициенты бикубического многочлена. Существуют также и C_y и C_z , которые определяют коэффициенты $y(s, t)$ и $z(s, t)$.

Рассмотрим представление поверхностей в форме Эрмита. К поверхностям в форме Эрмита наиболее удобно предложить подход, который позволил бы использовать управляющие точки и касательные векторы для определения коэффициентов бикубического многочлена.

Заменяем в уравнении кубической кривой Эрмита T на S :

$$x(s) = SM_hG_{hx}.$$

Перепишем это уравнение так, чтобы геометрическая матрица Эрмита была не константой, а функцией t :

$$x(s, t) = SM_hG_{hx}(t) = SM_h \begin{bmatrix} P_1(t) \\ P_4(t) \\ R_1(t) \\ R_4(t) \end{bmatrix}_x. \quad (10.1)$$

Функции $P_{1x}(t)$ и $P_{4x}(t)$ описывают x -компоненты начальной и конечной точек кривой, задаваемой параметром s . Для каждого значения t определяются некоторые две конечные точки. Аналогично $R_{1x}(t)$ и $R_{4x}(t)$ описывают касательные векторы в конечных точках кубической кривой, построенной в зависимости от s . На рисунке 10.1 показаны кривые $P_1(t)$ и $P_4(t)$, а также кубические кривые, для которых $t = 0, 0,2, 0,4, 0,8, 1,0$. Фрагмент поверхности можно представить как фигуру, построенную путём интерполяции между кривыми $P_1(t)$ и $P_4(t)$, при этом касательный вектор в начальной точке будет $R_1(t)$, а в конечной – $R_4(t)$. В том случае, если интерполируемые линии являются прямыми, получается линейчатая поверхность. Если к тому же кривые $P_1(t)$ и $P_4(t)$ лежат в одной плоскости, то линейчатая поверхность будет плоской, а кусок поверхности окажется четырёхсторонним многоугольником.

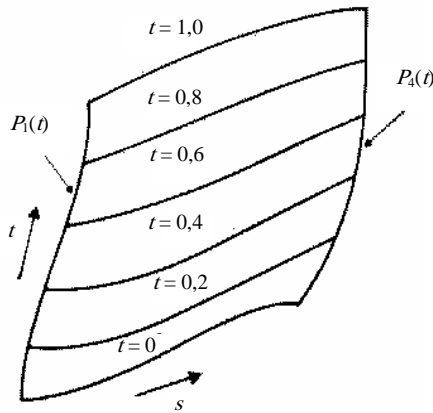


Рис. 10.1. Кривые постоянного значения параметра на бикубической поверхности: $P_1(t)$ при $s = 0$, $P_4(t)$ при $s = 1$

Пусть теперь каждая из кривых $P_1(t)$, $P_4(t)$, $R_1(t)$ и $R_4(t)$ представлена кубическим многочленом в форме Эрмита:

$$P_{1x}(t) = TM_h \begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \\ q_{14} \end{bmatrix}_x; \quad P_{4x}(t) = TM_h \begin{bmatrix} q_{21} \\ q_{22} \\ q_{23} \\ q_{24} \end{bmatrix}_x;$$

$$R_{1x}(t) = TM_h \begin{bmatrix} q_{31} \\ q_{32} \\ q_{33} \\ q_{34} \end{bmatrix}_x; \quad R_{4x}(t) = TM_h \begin{bmatrix} q_{41} \\ q_{42} \\ q_{43} \\ q_{44} \end{bmatrix}_x.$$

Четыре кубических многочлена можно представить в виде вектор строки

$$[P_1(t), P_4(t), R_1(t), R_4(t)] = TM_h = \begin{bmatrix} q_{11} & q_{21} & q_{31} & q_{41} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix}_x.$$

Транспонируя данное уравнение и используя тождество $(ABC)^T = C^T B^T A^T$, получим

$$\begin{bmatrix} P_1(t) \\ P_4(t) \\ R_1(t) \\ R_4(t) \end{bmatrix} = \begin{bmatrix} q_{11} & q_{21} & q_{31} & q_{41} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix}_x M_h^T T^T = Q_x M_h^T T^T. \quad (10.2)$$

Теперь подставим (10.2) в (10.1):

$$x(s, t) = SM_h Q_x M_h^T T^T.$$

Аналогично

$$y(s, t) = SM_h Q_y M_h^T T^T;$$

$$z(s, t) = SM_h Q_z M_h^T T^T.$$

Как определить Q_x , Q_y и Q_z с помощью точек и углов наклона? Из предшествующих выкладок следует, что коэффициент q_{11} есть $x(0, 0)$, так как является начальной точкой для $P_{1x}(t)$, которая в свою очередь задаёт начальную точку для $x(s, 0)$. Аналогично q_{12} есть $x(0, 1)$, так как является конечной точкой $P_{1x}(t)$, которая в свою очередь задаёт начальную точку для $x(s, 1)$. Видно также, что

$$q_{13} = \frac{dx}{dt}(0, 0),$$

так как является начальным касательным вектором для $P_{1x}(t)$

$$q_{33} = \frac{d^2 x}{ds dt}(0, 0),$$

поскольку представляет собой начальный касательный вектор для $R_{1x}(t)$, который, в свою очередь, задаёт начальный тангенс угла наклона для $x(s, 0)$.

Используя эти рассуждения, можно записать матрицу Q_x :

$$Q_x = \begin{bmatrix} x_{00} & x_{01} & \left(\frac{dx}{dt}\right)_{00} & \left(\frac{dx}{dt}\right)_{01} \\ x_{10} & x_{11} & \left(\frac{dx}{dt}\right)_{10} & \left(\frac{dx}{dt}\right)_{11} \\ \left(\frac{dx}{ds}\right)_{00} & \left(\frac{dx}{ds}\right)_{01} & \left(\frac{d^2 x}{ds dt}\right)_{00} & \left(\frac{d^2 x}{ds dt}\right)_{01} \\ \left(\frac{dx}{ds}\right)_{10} & \left(\frac{dx}{ds}\right)_{11} & \left(\frac{d^2 x}{ds dt}\right)_{10} & \left(\frac{d^2 x}{ds dt}\right)_{11} \end{bmatrix}.$$

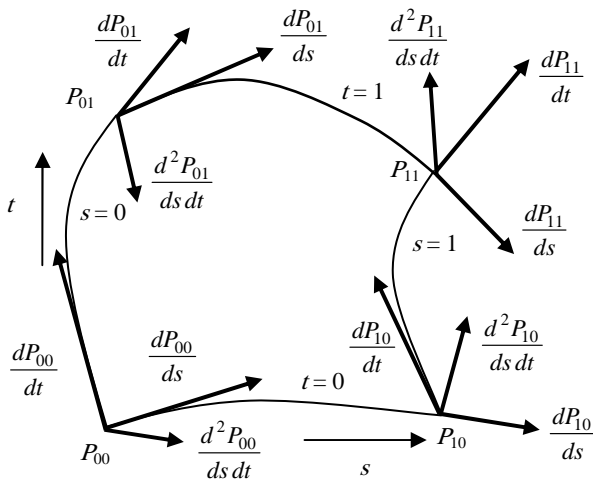


Рис. 10.2. Параметры для поверхности Эрмита

В верхней левой части размером 2×2 находятся четыре координаты углов куска поверхности, в верхней правой и нижней левой частях матрицы помещены тангенсы углов наклона касательных векторов в угловых точках для каждой из граничных параметрических кривых и, наконец, в нижней правой части расположены частные производные по обоим параметрам в угловых точках. Эти частные производные нередко называют кривизной, так как чем больше их значения, тем сильнее изгиб (как в спирали) в угловой точке куска поверхности. На рисунке 10.2 показан фрагмент, в угловых точках которого эти параметры написаны.

Отметим, что форма Эрмита для бикубических кусков поверхности является одной из форм куска Кунса, так как она широко использовалась Кунсом, пионером применения машинной графики в автоматизации проектирования. Их иногда ещё называют поверхностями Фергюссона, также в честь одного из первых разработчиков представлений поверхностей.

Аналогично тому, как кубический многочлен Эрмита обеспечивает C^1 -непрерывность при переходе от одного сегмента к другому, бикубический многочлен Эрмита позволяет достичь C^1 -непрерывности при переходе от одного куска поверхности к другому. Необходимые условия для этого состоят в том, чтобы кривые, заданные на общем ребре, были одинаковыми на каждом из кусков, а также, чтобы касательные векторы, пересекающие ребро, имели одно и то же направление (их длины могут быть разными). Если общее ребро строится при фиксированном значении параметра s (т.е. t изменяется от 0 до 1), в этом случае строки матриц кусков должны соответствовать друг другу. Из уравнений (10.5) – (10.8) следует, что первая строка Q описывает ребро куска при $s = 0$, вторая строка определяет ребро, соответствующее $s = 1$, третья – задаёт касательный вектор вдоль ребра $s = 0$ и четвёртая – определяет касательный вектор вдоль ребра $s = 1$.

11. МОДЕЛИ ОСВЕЩЕНИЯ

Световая энергия, падая на поверхность, может быть отражена, пропущена и поглощена. Облучаемый объект можно увидеть, только если он отражает или пропускает свет. Количество отражённой, пропущенной и поглощённой энергии зависит от длины волны света.

Свойства отражённого света зависят от направления и формы источника света, от ориентации и свойств поверхности. Различают зеркальное и диффузное отражение света от объекта. Диффузное отражение света происходит, когда свет как бы проникает под поверхность объекта, поглощается, а затем вновь испускается. При этом положение наблюдателя не имеет значения, так как диффузно отражённый свет рассеивается равномерно по всем направлениям. Зеркальное отражение происходит от внешней поверхности.

Математическое описание процесса отражения света от реальной поверхности довольно сложное. В компьютерной графике обычно используют упрощенные модели [22, 25].

Рассмотрим диффузное отражение. Угол между падающим лучом света и нормалью поверхности (θ) (рис. 11.1). Интенсивность отражённого света пропорциональна косинусу угла θ , согласно закона Ламберта:

$$I = I_l k_d \cos \theta, \quad 0 \leq \theta \leq \pi/2,$$

где I – интенсивность отраженного света; I_l – интенсивность точечного источника света; k_d – коэффициент диффузного отражения ($0 \leq k_d \leq 1$); θ – угол между вектором, направленным на источник света, и нормалью к поверхности.

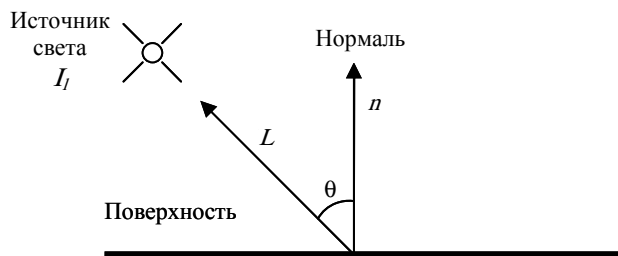


Рис. 11.1. Диффузное отражение

Поверхность предметов, изображённых при помощи простой диффузной модели освещения, выглядит блёклой и матовой. Объекты, на которые не падает свет, кажутся чёрными, что напоминает эффект от фотовспышки. Однако на объекты реальных сцен падает ещё и рассеянный свет, отражённый от окружающей обстановки, например от стен комнаты. В компьютерной графике рассеянное освещение учитывается как дополнительное слагаемое в модели освещения:

$$I = I_a k_a + I_l k_d \cos \theta, \quad 0 \leq \theta \leq \pi/2,$$

где I_a – интенсивность рассеянного света; k_a – коэффициент диффузного отражения рассеянного света ($0 \leq k_a \leq 1$).

Для большей реалистичности в модель освещения добавляют линейное затухание, связанное с конкретным расстоянием d от центра проекции до объекта.

$$I = I_a k_a + \frac{I_l k_d \cos \theta}{d + k}, \quad (11.1)$$

где k – произвольная постоянная.

Рассмотрим систему, состоящую из источника света и поверхности и наблюдателя (рис. 11.2). Если у поверхности отсутствуют какие-либо неровности, шероховатости, то такие поверхности считаются идеально зеркальными. Собственный цвет

таких поверхностей не наблюдается. Световая энергия падающего луча (вдоль вектора L) отражается только по линии отражения (вдоль вектора R). Рассеивание в стороны от линии отражения отсутствует. В реальной жизни такие поверхности не встречаются.

В простых моделях освещения обычно пользуются имперической моделью Буи-Туонга Фонга [22], так как физические свойства зеркального отражения очень сложны. Модель Фонга имеет вид

$$I = I_l k_s \cos^n \alpha,$$

где k_s – коэффициент зеркального отражения; n – степень, аппроксимирующая пространственное распределение зеркально отражённого света.

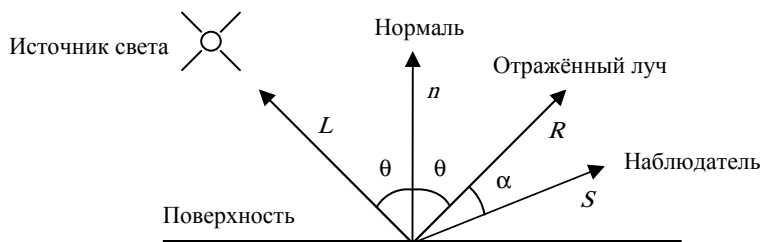


Рис. 11.2. Зеркальное отражение

Объединив диффузное и зеркальное отражение, получим

$$I = I_a k_a + \frac{I_l}{d+k} (k_d \cos \theta + k_s \cos^n \alpha).$$

Если имеются несколько источников света, то их эффекты суммируются:

$$I = I_a k_a + \sum_{j=1}^m \frac{I_{l_j}}{d+k} (k_d \cos \theta_j + k_s \cos^n_j \alpha_j), \quad (11.2)$$

где m – количество источников света.

Надо заметить, что все приведённые выше модели освещения определяются для каждой составляющей цвета, т.е. для R , G и B .

Если при построении полигональной поверхности для каждой грани используется по одной нормали, то модель освещения создаёт изображение, состоящее из отдельных многоугольников. Для плавной закраски полигональной поверхности используется метод Гуро [22, 23]. В данном алгоритме сначала определяется интенсивность освещения вершин по модели диффузного отражения, а затем с помощью билинейной фильтрации вычисляется интенсивность каждого пикселя на сканирующей строке.

Для вычисления освещенности вершин 3D объекта приходится вычислять нормаль в вершине, а это осуществляется через усреднение нормалей полигонов сходящихся в этой вершине. Простое суммирование нормалей полигонов обеспечивает получения нормали в вершине полигональной сетки. Следует отметить, что делить полученную сумму нормалей на их количество не обязательно, так как нас интересует не длина вектора, а его направление.

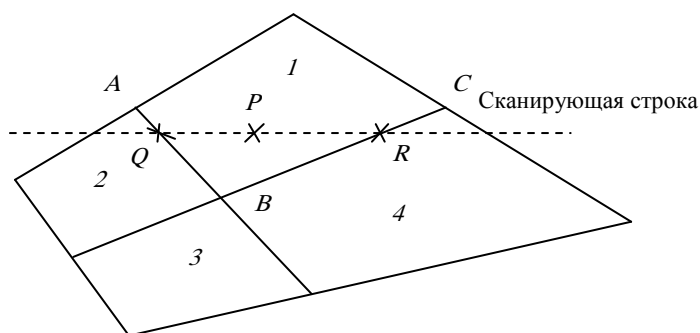


Рис. 11.3. Закраска полигонов методом Гуро

Можно вычислить обобщённую нормаль в вершине через усреднение векторного произведения всех рёбер полигональной сетки сходящихся в вершине.

Рассмотрим, например, участок полигональной поверхности на рис. 11.3. Значение интенсивности в точке P определяется линейной интерполяцией интенсивности в точках Q и R . Для получения интенсивности в точке Q , полученной в результате пересечения ребра полигона со сканирующей строкой, нужно линейной интерполяцией интенсивностей A и B найти

$$I_Q = u I_A + (1-u) I_B, \quad 0 \leq u \leq 1,$$

где $u = AQ/AB$.

Аналогично для получения интенсивности в точке R линейно интерполируются интенсивности в вершинах B и C , т.е.

$$I_R = wI_B + (1-w)I_C, \quad 0 \leq w \leq 1,$$

где $w = BR/BC$.

Наконец, линейной интерполяцией по строке между Q и R находится интенсивность освещений в точке P , т.е.

$$I_P = tI_Q + (1-t)I_R, \quad 0 \leq t \leq 1,$$

где $t = QP/QR$.

Закраска Гуро лучше всего выглядит в сочетании с простой моделью с диффузным отражением (11.2). Для более реалистической визуализации интерполяционной закраски поверхностей, заданных малым количеством полигонов применяется метод Фонга [22, 23]. Эта закраска требует больших вычислительных затрат, однако она позволяет разрешить многие проблемы метода Гуро.

При закраске Фонга интерполируются не интенсивности освещения, а нормали в каждой точке полигональной поверхности. Благодаря такому подходу, мы можем использовать более сложные модели освещения, как, например, модели, учитывающие зеркальную составляющую отражённого света (11.2). В результате можем получить блики на полигональной поверхности.

ЗАКЛЮЧЕНИЕ

В учебном пособии изложен материал, раскрывающий два важных аспекта любой САПР: передача информации в вычислительных системах и алгоритмы отображения имеющейся графической информации на экране. Данные системы являются наиболее динамично развивающимися и используются конечным пользователем в повседневной жизни.

Материалы, изложенные в работе, позволяют не только понять суть процесса передачи информации на удалённую ЭВМ и её отображения, но и использовать их при настройке и локальной обработке САПР.

СПИСОК ЛИТЕРАТУРЫ

1. Блэк, Ю. Сети ЭВМ. Протоколы, стандарты, интерфейсы / Ю. Блэк. – М. : Мир, 1990.
2. Браун, С. Операционная система UNIX / С. Браун. – М. : Мир, 1986. – 463 с.
3. Башлы, П.Н. Современные сетевые технологии : учебное пособие / П.Н. Башлы. – М. : Мир, 1997.
4. Мячев, А.А. Интерфейсы и сети ЭВМ : англ.-рус. толков. словарь / А.А. Мячев. – М. : Радио и связь, 1994.
5. Linux Programmer's Manual [Электронный ресурс] http://zed.karelia.ru/4them/local_mans.html.
6. Brian Hall. Beej's Guide To Unix IPC. : [Электронный ресурс] / Brian Hall. <http://www.ecst.csuchico.edu/~beej/guide/ipc/>.
7. Brian Hall. Beej's Guide to Network Programming. : [Электронный ресурс] / Brian Hall. <http://www.ecst.csuchico.edu/~beej/guide/net/>.
8. Sandra Loosemore with Richard M. Stallman, Roland McGrath, Andrew Oram, and Ulrich Drepper. The GNU C Library Reference Manual. Edition 0.09 Draft, last updated 28 August 1999, for Version 2.2 Beta. 1080 pages, 31 chapters and 6 appendices (2 volume set). http://www.gnu.org/software/libc/manual/html_node/index.html.
9. Tim Carstens. Programming with pcap : [Электронный ресурс] <http://www.tcpdump.org/pcap.htm>.
10. Кузнецов, С.Д. Операционная система UNIX : [Электронный ресурс] / С.Д. Кузнецов. http://fizmat.vspu.ru/citforum/operating_systems/unix/contents.shtml.
11. Хэвиленд, К. Системное программирование в UNIX. Руководство программиста по разработке ПО / К. Хэвиленд, Д. Грэй, Б. Салама ; пер. с англ. Р.В. Павлова; гл. ред. И.М. Захаров. – М. : ДМК Пресс, 2000. – 364 с.
12. Системные вызовы и библиотеки ОС UNIX : учебно-метод. комплекс / Институт операционных систем при МИЭТ. – М. : Научная книга, 1995.
13. Расширенное описание сетей UNIX : учебно-метод. комплекс / Институт операционных систем при МИЭТ. – М. : Научная книга, 1995. – 246 с.
14. ОС UNIX для программистов : учебно-метод. комплекс / Институт операционных систем при МИЭТ. – М. : Научная книга, 1995.
15. Свиридов, С.В. Системные вызовы ОС UNIX / С.В. Свиридов. – М. : Память, 1992.

16. Керниган, Брайан В. UNIX – универсальная среда программирования / пер.с англ. А.М. Березко, В.А. Иващенко ; под ред. и с предисл. М.И. Белякова. – М. : Финансы и статистика, 1992.
17. Блэк, У. Интернет: протоколы безопасности : учебный курс / У. Блэк ; пер. с англ. С. Нефедьева, А. Солоницыной. – СПб.-М.-Харьков-Минск : Питер, 2001.
18. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы : учебник / В.Г. Олифер. – СПб.-М.-Харьков-Минск : Питер, 1999.
19. Олифер, В.Г. Компьютерный сети: принципы, технологии, протоколы : учеб. пособие для студентов вузов, обучающихся по направлению «Информатика и вычислит. техника» / В.Г. Олифер. – СПб.-М.,-Харьков-Минск : Питер, 2001.
20. Олифер, В.Г. Новые технологии и оборудование IP-сетей / В.Г. Олифер, Н.А. Олифер ; гл. ред. Е. Кондукова. – Дюссельдорф-Киев-СПб. : ВНУ-Санкт-Петербург, 2001.
21. Фоли Дж., вэн Дэм А. Основы интерактивной машинной графики. В 2-х т. / Фоли Дж., вэн Дэм А. – М. : Мир, 1985.
22. Роджерс, Д. Алгоритмические основы машинной графики / Д. Роджерс. – М. : Мир, 1989.
23. Пореев, В.Н. Компьютерная графика / В.Н. Пореев. – СПб. : БХВ-Петербург, 2002.
24. Шикин, Е.В. Начала компьютерной графики / Е.В. Шикин, А.В. Боресков, А.А. Зайцев. – М. : ДИАЛОГ-МИФИ, 1993.
25. Шикин, Е.В. Компьютерная графика. Динамика, реалистические изображения / Е.В. Шикин, А.В. Боресков. – М. : ДИАЛОГ-МИФИ, 1995.
26. Шикин, Е.В. Компьютерная графика. Полигональные модели / Е.В. Шикин, А.В. Боресков. – М. : ДИАЛОГ-МИФИ, 2000.
27. Казанцев, А.В. Основы компьютерной графики / А.В. Казанцев – Казань : Изд-во КТУ, 2001.
28. Павлидис, Т. Алгоритмы машинной графики и обработки изображений / Т. Павлидис. – М. : Радио и связь, 1986.
29. Васильев, С.А. OpenGL. Компьютерная графика / С.А. Васильев. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2005.

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ	3
Часть I. ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ	4
1. ВВЕДЕНИЕ В ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ	4
2. КАНАЛЫ ПЕРЕДАЧИ ДАННЫХ	12
3. ЛОКАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ	19
4. СЕТЕВОЙ И ТРАНСПОРТНЫЙ УРОВНИ	25
Часть II. КОМПЬЮТЕРНАЯ ГРАФИКА	42
1. АЛГОРИТМ БРЕЗЕНХЕМА ДЛЯ ПЕРОБРАЗОВАНИЯ ОТРЕЗКОВ ПРЯМЫХ ИЗ ВЕКТОРНОЙ ФОРМЫ В РАСТРОВУЮ	42
2. АЛГОРИТМ БРЕЗЕНХЕМА ДЛЯ РАСТРОВОЙ РАЗВЁРТКИ ОКРУЖНОСТИ	44
3. АФФИННЫЕ ПРЕОБРАЗОВАНИЯ НАД ДВУМЕРНЫМИ ОБЪЕКТАМИ	45
4. ТРЁХМЕРНЫЕ АФФИННЫЕ ПРЕОБРАЗОВАНИЯ 3ДОБЪЕКТОВ	49
5. ПРОЕЦИРОВАНИЕ ТРЁХМЕРНЫХ ОБЪЕКТОВ	52
6. РАСТРОВОЕ ТЕКСТУРИРОВАНИЕ	55
7. АЛГОРИТМ «ПЛАВАЮЩЕГО ГОРИЗОНТА» ДЛЯ ВИЗУАЛИЗАЦИИ ФУНКЦИОНАЛЬНО-ЗАДАННЫХ ПОВЕРХНОСТЕЙ	57
8. АЛГОРИТМ УДАЛЕНИЯ НЕВИДИМЫХ ГРАНЕЙ, ИСПОЛЗУЮЩИЙ BSP-ДЕРЕВА	59
9. ПОСТРОЕНИЕ КРИВЫХ	63
10. ПОСТРОЕНИЕ ПАРАМЕТРИЧЕСКИХ КУБИЧЕСКИХ ПОВЕРХНОСТЕЙ	67
11. МОДЕЛИ ОСВЕЩЕНИЯ	72
ЗАКЛЮЧЕНИЕ	76

