Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»

Факультет вечернего, заочного и дистанционного обучения
Кафедра ЭВМ

Контрольная работа
по дисциплине «Вычислительные комплексы, системы и сети».
Вариант №1
студента 4 курса 500503 учебной группы
Авсеева С.П.

Минск 2009

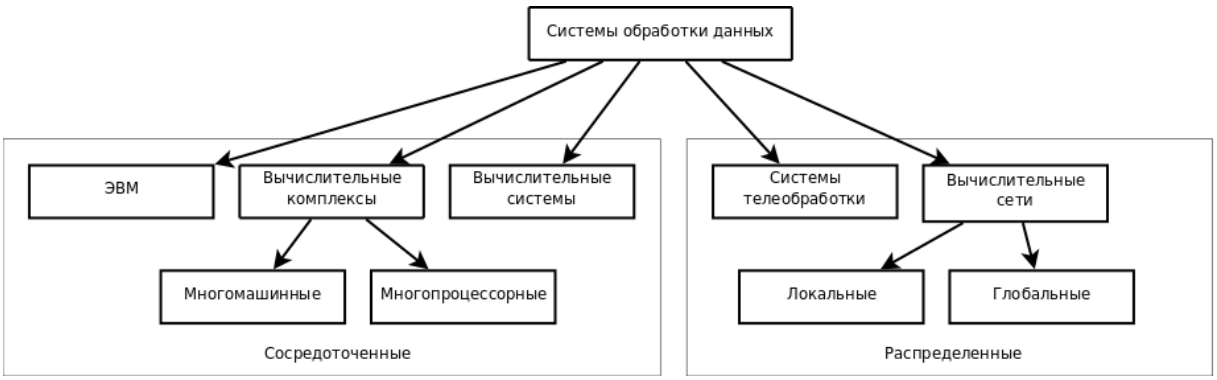
Содержание

1. Классификация и основные характеристики систем обработки данных

Система обработки данных (СОД) — совокупность технических средств и программного обеспечения, предназначенная для информационного обслуживания пользователей и технических объектов. В состав технических средств входит оборудование для ввода, хранения, преобразования и вывода данных, в том числе ЭВМ, устройства сопряжения ЭВМ с объектами, аппаратура передачи данных, и линии связь. Программное обеспечение (программные средства) — совокупность программ, реализующих возложенные на систему функции. Функции СОД состоят в выполнении требуемых актов обработки данных: ввода, хранения, преобразования и вывода.

1.1. Классификация

Классифицируются СОД в зависимости от способа построения. СОД, построенные на основе отдельных ЭВМ, вычислительных комплексов и систем, образуют класс сосредоточенных (централизованных) систем, в которых вся обработка реализуется ЭВМ, вычислительным комплексом или специализированной системой. Системы телеобработки и вычислительные сети относятся к классу распределенных систем, в которых процессы обработки данных рассредоточены по многим компонентам. При этом системы телеобработки считаются распределенными в некоторой степени условно, поскольку основные функции обработки данных здесь реализуются централизованно — в одной ЭВМ или вычислительном комплексе.



Сосредоточенными СОД называют такие системы, в которых оборудование размещено на небольшом расстоянии и связи между ними реализу-

ются внутренними интерфейсами.

Распределёнными СОД называют системы, в которых элементы находятся на значительном удалении друг от друга и связи между ними обеспечиваются каналами передачи данных.

Вычислительным комплексом называется СОД, включающая две и более машины или два и более процессора, работающих под управлением общего ПО.

СОД, ориентированные на решение ограниченного определённого круга задач, называются вычислительной системой. Ориентация может выполняться программно или аппаратно.

Под системой телеобработки понимают СОД, в которой источники и приёмники информации, включая термины пользователей, находятся на значительном удалении от обрабатывающего объекта и связи с ним обеспечиваются каналами передачи данных.

Вычислительной сетью называют СОД, в которой ЭВМ и комплексы, а также некоторое терминальное оборудование находится на значительном удалении друг от друга, и связи между ними осуществляются по каналам передачи данных.

В зависимости от территории, которые занимают сети, они делятся на глобальные — расстояние достигает сотен км; региональные — территория города, района, расстояние между машинами десятки км; локальные — ЭВМ на расстоянии десятков метров.

По режиму работы СОД делятся на однопрограммные и мультипрограммные.

По особенностям функционирования во времени – СОД, работающие в реальном масштабе времени и СОД, работающие в нереальном масштабе времени.

По режиму обслуживания пользователей делятся на:

1. СОД индивидуального пользования;
2. СОД с пакетной обработкой;
3. СОД коллективного использования.

1.2. Основные характеристики СОД

Основными характеристиками СОД являются производительность, время ответа, надежность и стоимость. В дополнение к ним используются следующие характеристики: габариты, масса, потребляемая мощность, диапазон рабочих температур, ремонтопригодность и др.

1.2.1. Производительность

Производительность — характеристика вычислительной мощности системы, определяющая количество вычислительной работы, выполняемой системой за единицу времени.

Производительность СОД определяется номинальной комплексной системой и производительностью на рабочей нагрузке.

Номинальная производительность определяется быстродействием устройств, входящих в СОД.

$$V = (V_1, V_2, \dots V_n)$$

С целью быстрого сравнения различных систем по номинальной производительности быстродействие параллельно работающих устройств представляет суммарное быстродействие. Устройство, не влияющее на решение задачи, исключают из решения.

$$V = \sum V_i$$

Для более предметного сравнения вводят понятие комплексной производительности. Она определяется набором быстродействия устройств при решении системной задачи.

$$V^* = (V_1^*, V_2^*, \dots, V_n^*) \text{ , где } V_i^* < V_i.$$

Системная производительность учитывает влияние СПО на время решения задачи. Для её оценки используется коэффициент загрузки устройства.

$$\rho_i = \frac{T_i}{T}, \textit{где}$$

T_i — время работы устройства СОД;
T — общее время работы СОД.

$$V_1^{**} = \rho_1 \cdot V_1, V_2^{**} = \rho_2 \cdot V_2, \ldots, V_n^{**} = \rho_n \cdot V_n . V^{**} = (V_1^{**}, V_2^{**}, \ldots, V_n^{**})$$

Для систем, находящихся в эксплуатации и решающих определённые задачи, системную производительность определяют как производительность на рабочей нагрузке.

$$\lambda = \frac{n}{T} \text{ [задач/час]}$$

Производительность на рабочей нагрузке можно определить через средний интервал времени между моментами окончания решения соответствующих задач.

1.2.2. Время ответа

Время ответа, иначе время пребывания заданий, (задач) в системе, — длительность промежутка времени от момента поступления задания в систему до момента окончания его выполнения.

В общем случае время ответа — случайная величина, что обусловлено следующими факторами:

1. влиянием исходных данных на число операций ввода, обработки и вывода данных и непредсказуемостью значений исходных данных;

2. влиянием состава смеси задач, одновременно находящихся в системе, и непредсказуемостью состава смеси из-за случайности момента поступления задач на обработку.

Время ответа как случайная величина наиболее полно характеризуется функцией распределения $P(u < x)$ или функцией плотности вероятностей $p(u)$. Чаще всего время ответа оценивается средним значением, которое определяется как статистическое среднее случайной величины $u_i, i = 1, \ldots, n$, наблюдаемой для задач J_i :

$$U = \frac{1}{n} \sum_{i=1}^{n} u_i; \; U_{\text{вып.}} = U_{\text{реш.}} + U_{\text{ожид.}}$$

1.2.3. Характеристики надежности

Надежность — свойство системы выполнять возложенные на нее функции в заданных условиях функционирования с заданными показателями качества: достоверностью результатов, пропускной способностью, временем ответа и др. Работоспособность системы или отдельных ее частей нарушается из-за отказов аппаратуры — выхода из строя элементов или соединений.

Важнейшая характеристика надежности – интенсивность отказов, определяющая среднее число отказов за единицу времени, как правило, за один час. Интенсивность отказов зависит от числа элементов и соединений, составляющих систему. Если любой отказ носит катастрофический характер, т. е. приводит к нарушению работоспособности системы, то интенсивность отказов в систем

$$\lambda_0 = \sum_{i=1}^{n} \lambda_i \text{ , где}$$

λ_i — интенсивность отказов i-го элемента.

Надежность системы может быть повышена за счет резервирования ее элементов — дублирования, троирования и т.д. Однако резервирование приводит к существенному увеличению стоимости системы.

1.2.4. Стоимость

Стоимость СОД – это суммарная стоимость технических средств и программного обеспечения. Стоимость технических средств определяется их составом и техническими характеристиками.

Стоимость СОД влияет на стоимость решения задачи, которая определяется стоимостью ресурсов используемых задачей:

$$S = \sum_{i=1}^{N} c_i \Theta_i \text{ , где}$$

c_i — стоимостной коэффициент, определяющий стоимость использования единицы ресурса i (миллиона процессорных операций, килобайта памяти и др.), и Θ_i — объем ресурса, используемый задачей.

2. Распределение основных функций управления по системам сети

В техническом отношении КС рассматривают, как совокупность систем, связанных между собой некоторой передающей средой.

В качестве систем в сети выступают главные и терминальные ЭВМ, сетевые адаптеры, коммутаторы, концентраторы, мосты и маршрутизаторы.

В каждой из этих систем существует некоторая совокупность процессов. Процессы этих систем должны взаимодействовать между собой. Для обеспечения гибкости, открытости и эффективности сетей управления процессами в ней осуществляется по многоуровневой схеме. Число уровней управления и распределение функций между ними влияет на сложность ПО и эффективность функционирования сети.

МОС (ISO) предложило семиуровневую модель управления. На каждом уровне определены функции, которые тесно связаны между собой и мало связаны с соседними. Связь между уровнями управления реализуется стандартным путем.

1. Физический

   На физический уровень управления возлагается коммутация канала, передача данных, разъединение канала. Ввиду большой протяженности канала связи возможно искажение бит сообщения, вероятность ошибок $p = 10^{-4} \div 10^{-6}$ ошибок/бит.

2. Канальный

   В задачу канального уровня управления входит повышение достоверности передачи данных, формирование кадров, при необходимости выполнение процедуры битстаффинга. Вероятность не обнаружения ошибки $p = 10^{-8} \div 10^{-12}$

3. Сетевой

   На сетевом уровне осуществляется маршрутизация пакетов по сети, т.е. выбирается направление передачи.

4. Транспортный

   Транспортный уровень является уровнем сопряжения сети передачи

данных с протоколами выше расположенных уровней. На этом уровне управления из сообщений формируются пакеты, проверяется наличие вызываемого абонента, процесса, с которым организуется взаимодействие, формируются блоки связи, необходимые для взаимодействия удаленных процессов.

5. Сеансовый

   На сеансовом уровне организуются порты взаимодействия, устанавливаются сеансы связи, чтобы осуществлять распределенное решение сложныз задач несколькими ЭВМ сети.

6. Представительный

   На представительном уровне унифицируются команды и процедуры ОС, терминалов, форма представления чисел и т.д.

7. Прикладной

   На прикладном уровне обеспечивается выполнение заданий пользователя сети. Организуются программные средства, организующие управление, удаленный ввод заданий на управление сервером, электронная почта, работа с файлами и т.д.

3. Причины возникновения ошибок передачи в сетях. Контроль достоверности передачи данных в сетях передатчиком

Канальный уровень должен обнаруживать ошибки передачи данных, связанные с искажением бит в принятом кадре данных или с потерей кадра, и по возможности их корректировать.

Большая часть протоколов канального уровня выполняет только первую задачу — обнаружение ошибок, считая, что корректировать ошибки, то есть повторно передавать данные, содержавшие искаженную информацию, должны протоколы верхних уровней. Так работают такие популярные протоколы локальных сетей, как Ethernet, Token Ring, FDDI и другие. Однако существуют протоколы канального уровня, например LLC2 или LAP-B, которые самостоятельно решают задачу восстановления искаженных или потерянных кадров.

Очевидно, что протоколы должны работать наиболее эффективно в типичных условиях работы сети. Поэтому для сетей, в которых искажения и потери кадров являются очень редкими событиями, разрабатываются протоколы типа Ethernet, в которых не предусматриваются процедуры устранения ошибок. Действительно, наличие процедур восстановления данных потребовало бы от конечных узлов дополнительных вычислительных затрат, которые в условиях надежной работы сети являлись бы избыточными.

Напротив, если в сети искажения и потери случаются часто, то желательно уже на канальном уровне использовать протокол с коррекцией ошибок, а не оставлять эту работу протоколам верхних уровней. Протоколы верхних уровней, например транспортного или прикладного, работая с большими тайм-аутами, восстановят потерянные данные с большой задержкой. В глобальных сетях первых поколений, например сетях X.25, которые работали через ненадежные каналы связи, протоколы канального уровня всегда выполняли процедуры восстановления потерянных и искаженных кадров.

Поэтому нельзя считать, что один протокол лучше другого потому, что он восстанавливает ошибочные кадры, а другой протокол - нет. Каждый

протокол должен работать в тех условиях, для которых он разработан.

Все методы обнаружения ошибок основаны на передаче в составе кадра данных служебной избыточной информации, по которой можно судить с некоторой степенью вероятности о достоверности принятых данных. Эту служебную информацию принято называть контрольной суммой или (последовательностью контроля кадра - Frame Check Sequence, FCS). Контрольная сумма вычисляется как функция от основной информации, причем необязательно только путем суммирования. Принимающая сторона повторно вычисляет контрольную сумму кадра по известному алгоритму и в случае ее совпадения с контрольной суммой, вычисленной передающей стороной, делает вывод о том, что данные были переданы через сеть корректно.

Существует несколько распространенных алгоритмов вычисления контрольной суммы, отличающихся вычислительной сложностью и способностью обнаруживать ошибки в данных.

Контроль по паритету представляет собой наиболее простой метод контроля данных. В то же время это наименее мощный алгоритм контроля, так как с его помощью можно обнаружить только одиночные ошибки в проверяемых данных. Метод заключается в суммировании по модулю 2 всех бит контролируемой информации. Например, для данных 100101011 результатом контрольного суммирования будет значение 1. Результат суммирования также представляет собой один бит данных, который пересылается вместе с контролируемой информацией. При искажении при пересылке любого одного бита исходных данных (или контрольного разряда) результат суммирования будет отличаться от принятого контрольного разряда, что говорит об ошибке. Однако двойная ошибка, например 110101010, будет неверно принята за корректные данные. Поэтому контроль по паритету применяется к небольшим порциям данных, как правило, к каждому байту, что дает коэффициент избыточности для этого метода 1/8. Метод редко применяется в вычислительных сетях из-за его большой избыточности и невысоких диагностических способностей.

Вертикальный и горизонтальный контроль по паритету представляет собой модификацию описанного выше метода. Его отличие состоит в том, что исходные данные рассматриваются в виде матрицы, строки которой

составляют байты данных. Контрольный разряд подсчитывается отдельно для каждой строки и для каждого столбца матрицы. Этот метод обнаруживает большую часть двойных ошибок, однако обладает еще большей избыточностью. На практике сейчас также почти не применяется.

Циклический избыточный контроль (Cyclic Redundancy Check, CRC) является в настоящее время наиболее популярным методом контроля в вычислительных сетях (и не только в сетях, например, этот метод широко применяется при записи данных на диски и дискеты). Метод основан на рассмотрении исходных данных в виде одного многоразрядного двоичного числа. Например, кадр стандарта Ethernet, состоящий из 1024 байт, будет рассматриваться как одно число, состоящее из 8192 бит. В качестве контрольной информации рассматривается остаток от деления этого числа на известный делитель R. Обычно в качестве делителя выбирается семнадцати- или тридцати трехразрядное число, чтобы остаток от деления имел длину 16 разрядов (2 байт) или 32 разряда (4 байт). При получении кадра данных снова вычисляется остаток от деления на тот же делитель R, но при этом к данным кадра добавляется и содержащаяся в нем контрольная сумма. Если остаток от деления на R равен нулю1 (1 Существуетнесколько модифицированная процедура вычисления остатка, приводящая к получению в случае отсутствия ошибок известного ненулевого остатка, что является более надежным показателем корректности.), то делается вывод об отсутствии ошибок в полученном кадре, в противном случае кадр считается искаженным.

Этот метод обладает более высокой вычислительной сложностью, но его диагностические возможности гораздо выше, чем у методов контроля по паритету. Метод CRC обнаруживает все одиночные ошибки, двойные ошибки и ошибки в нечетном числе бит. Метод обладает также невысокой степенью избыточности. Например, для кадра Ethernet размером в 1024 байт контрольная информация длиной в 4 байт составляет только 0,4 %.

4. Назначение протокола HDLC. Типы, форматы и назначение кадров протокола HDLC

High-Level Data Link Control (HDLC) — бит-ориентированный протокол высокоуровнего управления каналом передачи данных, является опубликованным ISO стандартом и базовым для построения других протоколов канального уровня (SDLC, LAP, LAPB, LAPD, LAPX и LLC). Он реализует механизм управления потоком посредством непрерывного ARQ (скользящее окно) и имеет необязательные возможности (опции), поддерживающие полудуплексную и полнодуплексную передачу, одноточечную и многоточечную конфигурации, а так же коммутируемые и некоммутируемые каналы.

Кадры HDLC можно передавать, используя синхронные и асинхронные соединения. В самих соединениях нет механизмов определения начала и конца кадра, для этих целей используется уникальная в пределах протокола флаговая последовательность (FD — Frame Delimiter) '01111110'(0x7E в шестнадцатеричном представлении), помещаемая в начало и конец каждого кадра. Уникальность флага гарантируется использованием битстаффинга в синхронных соединениях и байтстаффинга в асинхронных. Битстаффинг — вставка битов, здесь — бита 0 после 5 подряд идущих битов 1. В байтстаффинге используется escape-последовательность, здесь — '01111101'(0x7D в шестнадцатеричном представлении), то есть байт FD(0x7E) в середине кадра заменяется последовательностью байтов (0x7D, 0x7E), а байт (0x7D) — последовательностью байтов (0x7D, 0x7D).

Во время простоя среды передачи при синхронном соединении FD постоянно передаётся по каналу для поддержания битовой синхронизации. Может иметь место совмещение последнего бита 0 одного флага и начального бита 0 следующего. Время простоя также называется межкадровым временным заполнением.

Структура кадров

Структура кадра HDLC, включая флаги FD:

Флаг	Адрес	Управляющее поле	Информационное поле	FCS	Флаг
8 бит	8 бит	8 или 16 бит	0 или более бит, кратно 8	16 бит	8 бит

Флаг конца одного кадра может(но не обязательно) быть начальным флагом следующего кадра.

Поле FCS(Frame Check Sequence) — контрольная последовательность, необходимая для обнаружения ошибок передачи. Её вычисление в основном производится методом циклического кодирования с производящим полиномом X16+X12+X5+1 (CRC-16) в соответствии с рекомендацией CCITT V.41. Это позволяет обнаруживать всевозможные кортежи ошибок длиной до 16 бит вызываемые одиночной ошибкой, а также 99,9984 % всевозможных более длинных кортежей ошибок. FCS составляется по полям Адрес, Управляющее поле, Информационное поле. В редких случаях используются другие методы циклического кодирования.

После просчета FCS на стороне приёмника он отвечает положительной или отрицательной квитанцией. Повтор кадра передающей стороной выполнятся по приходу отрицательной квитанции или по истечению таймаута.

Формат управляющего поля кадра HDLC

1	2	3	4	5	6	7	8	Разряды
0	N(S)			P/F	N(R)			I-формат
1	0	S-коды		P/F	N(R)			S-формат
1	1	U-коды		P/F	U-коды			U-формат

N(S) - порядковый номер передаваемого кадра, N(R) - порядковый номер принимаемого кадра, P/F - бит опроса/окончания.

Адресное поле определяет первичную или вторичную станции, участвующие в передаче конкретного кадра. Каждой станции присваивается уникальный адрес. В несбалансированной системе адресные поля в командах и ответах содержат адрес вторичной станции. В сбалансированных конфигурациях командный кадр содержит адрес получателя, а кадр ответа содержит адрес передающей станции.

Управляющее поле задает тип команды или ответа, а так же порядковые номера, используемые для отчетности о прохождении данных в канале между первичной и вторичной станциями. Формат и содержание управляющего поля определяют кадры трех типов: информационные (I), супервизорные (S) и ненумерованные (U).

- Информационный формат(I - формат) используется для передачи данных конечных пользователей между двумя станциями.

- Супервизорный формат (S - формат) выполняет управляющие функции: подтверждение (квитирование) кадров, запрос на повторную передачу кадров и запрос на временную задержку передачи кадров. Фактическое использование супервизорного кадра зависит от режима работы станции (режим нормального ответа, асинхронный сбалансированный режим, асинхронный режим ответа).

- Ненумерованный формат (U - формат) также используется для целей управления: инициализации или разъединения, тестирования, сброса и идентификации станции и т.д. Конкретный тип команды и ответа зависит от класса процедуры HDLC.

5. Протоколы транспортного уровня стандарта МОС (ISO)

Функции транспортного протокола двояки. С одной стороны, транспортный протокол определяет средства, необходимые для взаимодействия процессов, т.е. вводит интерфейс. С другой стороны, этот протокол организует сопряжение процессов, т.е. соединение компьютера с СПД, построенной по правилам, определяемым сетевым протоколом.

Транспортные модули, реализуемые в каждом из компьютеров, должны определить, какая связь должна быть реализована: внутрисистемная или межсетевая, распределить ресурсы между взаимодействующими процессами, установить соединение с удалённым процессом.

Транспортные модули должны устанавливать как глобальные, так и локальные связи. Функционирование транспортных модулей, как и в целом транспортного интерфейса, определяется транспортным протоколом. Этот протокол вводит набор процедур, реализуемый в транспортном интерфейсе, связанный с установлением соединения, формированием пакетов из сообщений, восстановление потерянных сообщений, а также создание портов взаимодействия. На этот же уровень управления могут быть возложены функции веления приоритетов, засекречивания данных и т.д.

Предусмотрены 18 процедур, которые инициализируются соответствующими командами или ответами(примитивами).

Все процедуры делятся на 5 каналов:

1. Установление соединения и разъединения

    1.1. ожидание – подготовка к соединению;

    1.2. соединение – установление соединения;

    1.3. согласие – согласие на установления соединения;

    1.4. отказ – отказ в установлении соединения;

    1.5. отмена – отмена режима ожидания;

    1.6. разъединение – разъединение соединения;

    1.7. разрыв – неуправляемое разъединение соединения.

2. Передача данных

    2.1. передача – передача сообщения для удалённого процесса;

    2.2. приём – готовность к приёму сообщения;

    2.3. отмена – отмена процедуры передачи;

    2.4. отмена приёма – отмена готовности к приёму.

3. Синхронизация

    3.1. приём прерывания – готовность к приёму прерываний;

    3.2. прерывания – передача прерывания;

    3.3. отмена прерывания – освобождение ресурсов после прерываний;

    3.4. рестарт(сброс) – повторный старт.

4. Датаграммная служба

    4.1. передача датаграмм – передача информации в датаграммном режиме;

    4.2. приём датаграмм – приём в датаграммном режиме.

5. Переключение

    5.1. переключение – изменение адреса местного процесса.

Примитивом ожидания процессы извещают транспортные модули о своей готовности приёму вызовов этим сигналом (процедурой) устанавливается путь между процессом и транспортным модулем, информируется транспортный модуль о готовности продела образовать соединение с удалённым процессом.

Процесс, выдавший примитив ожидания, становится доступным во всей сети. Для ликвидации режима ожидания используют примитив отмена. Этой процедурой разрывается связь процесса с транспортным модулем, процесс становится недоступным для взаимодействия с другими процессами сети.

Командой соединение инициализируется процедура, в соответствии с которой устанавливается связь с удалённым процессом. Для установления

соединения процесс выдаёт в транспортный модуль адрес выдаваемого абонента и требуемые транспортные услуги. Если соединение будет установлено, вызываемый абонент отвечает на это ответом согласие. Транспортный модуль, получив такой ответ, извещает вызывающего абонента о том, что связь установлена.

Командами приём и передача в соответствующих транспортных модулях устанавливаются буферы, через которые организуется взаимодействие между удалёнными процессами. На приёмной стороне процесс извещается о приёме сообщения после принятия всех пакетов данного сообщения. Согласование процедур приёма и передачи организуется через окно интерфейса X.25 может осуществляется путём квитирования, подтверждая каждую передачу командой приём.

Процедурой отмена передачи ликвидируется процедура передачи. Если сообщение не было передано полностью, оно уничтожается и об этом извещается удалённый транспортный модуль.

Командой отмена приёма отменяется приём сообщения, об этом извещается передающий транспортный модуль. Сообщение теряется.

Командой разъединение ликвидируется упорядоченным образом соединение между процессами. Процедура инициируется после окончания передачи всего сообщения. Удалённый транспортный модуль выдаёт эту команду после приема всего сообщения. Освобождаются все ресурсы.

Командой разрыв в экстренном порядке разъединяется соединение, при этом сообщения, которые были в сети, теряются.

Для передачи срочных данных используется процедура передача прерываний. В пакете передается причина передачи прерываний. Противоположный транспортный модуль извещает о приёме готовности ответом приём прерываний. Для приёма прерываний выделяется дополнительные ресурсы, которые в дальнейшем могут быть освобождены процедурой отмена прерываний.

Процедурой рестарт или сброс осуществляется возврат к исходному состоянию в случае нарушения синхронизации пот сообщению.

Для передачи и приёма одиночных сообщений служат процедуры:
Передача датаграмм = соединение + передача +разъединение.
Приём датаграмм = ожидание + согласие + приём + разъединение.

Процедура переключение используется для информирования транспортного модуля об изменении порта взаимодействия у данного процесса.

6. Протокол административного управления сетью. Защита данных и идентификация пользователей в сетях

6.1. Протокол административного управления сетью

Реализуется как распределённая система, модули которой формируются в клиентских станциях и серверах. Административная служба должна обеспечить:

1. обслуживание операторов в сети;
2. управление конфигурацией сети;
3. контроль состояние сети;
4. организация технического обслуживания;
5. управление режимами функционирования;
6. учёт использования ресурсов и сбор статистики.

Обслуживание операторов в сети сводится к подключению и отключению абонентов сети, обеспечению доступа операторов к элементам сети, получению информации о состоянии сети, контроль работоспособности и диагностики сети, сбор статистики о работе сети.

Управление конфигурацией сети сводится к подключению и отключению абонентов сети, отключению каналов и узлов связи.

Контроль состояния сети производится путём запросов у систем сети данных об именах и адресах активизированных процессов, данных о сессиях, логических каналов, данных о загрузке ресурсов и т.д.

Технические обслуживание сводится к наблюдению за состоянием сети, работоспособности компонентов, регистрации отказов и рестартов, регистрации отказов и рестартов в канале и т.д. Для этого часто используют эхо пакеты.

Управление режимом функционирования сети сводится к переадресации отдельных задач на другие компьютеры, выбору пропускной способности и режимов работы узлов связи, модернизации карт маршрутов, переадресацию числа разрешений на ввод пакетов в сеть. Для начисления

денежных затрат на решение задачи организуется учёт использования ресурсов в сети отдельными задачами.

6.2. Защита данных и идентификация пользователей в сетях

Защита информации в сетях может быть организована элементами экранизации или с использованием специальных программных средств защиты информации.

Наиболее уязвимы кабельные соединения. Для кодирования информации применяют методы криптографии, включающие шифрование и дешифрование передаваемой и хранимой информации. Алгоритмы известны, ключи для кодирования засекречиваются. Засекречиванию подлежат списки абонентов сетей, чтобы не допустить использование ресурсов сети не абонентами данной сети.

Идентификация пользователей осуществляется с помощью паролей. Пароли могут быть алфавитно–цифровые или на специальных карточках размещают пароль пользователя.