

УЗБЕКСКОЕ АГЕНСТВО СВЯЗИ И ИНФОРМАТИЗАЦИИ  
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

К защите допустить  
Зав. Кафедрой  
«С и СПД»  
Амирсаидов У.Б.  
\_\_\_\_\_ 2012 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

на тему **МОДЕЛИРОВАНИЕ ПРОТОКОЛА ПЕРЕДАЧИ  
ДАННЫХ HDLC**

Выпускник	_____	<u>Абдуллаев А.Н.</u>
	подпись	Ф.И.О.
Руководитель	_____	<u>Нурматова С.Б.</u>
	подпись	Ф.И.О.
Рецензент	_____	<u>Кадиров А.А.</u>
	подпись	Ф.И.О.
БЖД	_____	_____
	подпись	Ф.И.О.

Ташкент - 2012

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. СЕТИ И КАНАЛЫ ПЕРЕДАЧИ ИНФОРМАЦИИ.....	6
1.1 Характеристики и классификация информационных сетей.....	6
1.2 Многоуровневая архитектура информационных сетей.....	8
1.3 Разновидности каналов связи.....	17
ВЫВОДЫ.....	20
2. ОБЗОР ПРОТОКОЛА HDLC.....	21
2.1 Понятие протокола канального уровня.....	21
2.2 Типы, логические состояния и режимы работы станций.....	22
2.3 Управление потоком.....	26
2.4	Формат кадра
HDLC.....	27
2.5 Управляющее поле HDLC.....	31
2.6 Процессы передачи в протоколе HDLC.....	33
2.7 Подмножества HDLC.....	41
ВЫВОДЫ.....	47
3. ИММИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОТОКОЛА HDLC.....	48
3.1 Выбор языка программирования.....	48
3.2. Описание команд и ответов.....	49
3.3 Описание алгоритма.....	56
3.4 Системные параметры T1, N2, N1, K и рекомендации по их установке...59	
3.5 Листинг программы.....	60
ВЫВОДЫ.....	68
4. БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ.....	69
4.1 Гиподинамия и их влияние на организм человека.....	69
4.2. Пожарная безопасность на предприятиях.....	71
ЗАКЛЮЧЕНИЕ.....	78
ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА.....	80
ПРИЛОЖЕНИЯ.....	81

## ВВЕДЕНИЕ

Повсеместная разработка и внедрение новых информационных и инновационных технологий во все сферы и отрасли жизнедеятельности и рост потребностей в комплексной автоматизации организаций, предприятия и фирм обуславливает резкое возрастание объемов работ по созданию и внедрению систем обработки данных к качеству и эффективности которых предъявляются все более высокие требования. Разработка формализованных моделей и методов оптимального синтеза программного и информационного обеспечения модульных систем обработки данных, автоматизация технического проектирования оптимальных по заданным критериям систем обработки данных значительно повышает эффективность и качество создаваемых систем.

В качестве стандарта для протоколов канального уровня организацией ISO рекомендуется протокол HDLC (*High Level Data Link Control*). Он получил в мире телекоммуникаций чрезвычайно широкое распространение. На основе протокола HDLC разработано множество других, являющихся по своей сути некоторой адаптацией и упрощением ряда его возможностей по отношению к конкретной области применения. К такому подмножеству HDLC относятся часто используемые протоколы SDLC (*Synchronous Data Link Control*), LAP (*Link Access Procedure*), LAPB (*Link Access Procedure Balanced*), LAPD (*Link Access Procedure D-channel*), LAPM (*Link Access Procedure for Modems*), LLC (*Logical Link Network*), LAPX (*Link Access Procedure eXtention*) и ряд других. Например, протоколы LAPB и LAPD применяются в цифровых сетях ISDN (*Integrated Services Digital Network*), LAPM является базовым для стандарта коррекции ошибок V.42, LAPX является полудуплексным вариантом HDLC и используется в терминальных сетях и системах, работающих в стандарте Teletex, а протокол LLC (*Link Logic Control*) реализован практически во всех сетях с множественным доступом (например, в беспроводных локальных сетях).

### 3. СЕТИ И КАНАЛЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

#### 1.1 Характеристики и классификация информационных сетей

Современные телекоммуникационные технологии основаны на использовании информационных сетей.

*Коммуникационная сеть* - 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. При функциональном проектировании сетей решаются задачи синтеза топологии, распределения информации по узлам сети, а при конструкторском проектировании выполняются размещение пунктов в пространстве и проведение (трассировка) соединений.

*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 приемниками 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* - охватывающие значительное географическое пространство; среди территориальных сетей можно выделить сети региональные и глобальные, имеющие соответственно региональные или глобальные масштабы; региональные сети иногда называют сетями MAN (Metropolitan Area Network), а общее англоязычное название для территориальных сетей - WAN (Wide Area Network); *-локальные* (ЛВС) - охватывающие ограниченную территорию (обычно в пределах удаленности станций не более чем на несколько десятков или сотен метров друг от друга, реже на 1...2 км); локальные сети обозначают LAN (Local Area Network); *-корпоративные (масштаба предприятия)* - совокупность связанных между собой ЛВС, охватывающих территорию, на которой размещено одно предприятие или учреждение в одном или нескольких близко расположенных зданиях. Локальные и корпоративные вычислительные сети - основной вид вычислительных сетей, используемых в системах автоматизированного проектирования (САПР).

Особо выделяют единственную в своем роде 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, не интегрированные сети и подсети. *Интегрированная вычислительная сеть (интерсеть)* представляет собой взаимосвязанную совокупность многих вычислительных сетей, которые в интерсети называются подсетями.

В автоматизированных системах крупных предприятий подсети включают вычислительные средства отдельных проектных 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 видеоинформации, цифровых данных и т.п., и в этом случае они называются *сетями интегрального обслуживания*. Развитие интерсетей заключается в разработке средств сопряжения разнородных подсетей и стандартов для построения подсетей, изначально приспособленных к сопряжению. Подсети в интерсетях объединяются в соответствии с выбранной топологией с помощью *блоков взаимодействия*.

## **1.2 Многоуровневая архитектура информационных сетей**

Первые информационные сети были телефонные. Но с изобретением компьютера в телефонах стали использовать элементы ЭВМ (память, компьютерный интеллект), а в вычислительной технике поняли важность построения сетей, давно применявшихся для телефонной связи. Конечная цель всех этих нововведений – доставка информации любому корреспонденту по требуемому адресу и в надлежащее время.

В течение длительного периода процесс развития связи ЭВМ шел по пути создания и применения систем передачи данных по телефонным сетям общего пользования. Лишь когда обмен цифровой информацией достиг внушительных размеров, экономически целесообразным оказалось сооружение специализированных сетей передачи данных с коммутацией каналов и коммутацией пакетов.

В настоящее время по всему миру развернуты тысячи таких сетей, предоставляющие своим пользователям возможности связываться друг с другом. Размеры таких сетей простираются от небольших систем, соединяющих терминалы передачи данных в пределах отдельного здания или комплекса (например, промышленное предприятие), до больших географически распределенных сетей, охватывающих целые страны и даже весь земной шар.

В некоторых сетях применяется техника коммутации пакетов. В таких сетях от источника к получателю передаются блоки данных, называемые пакетами. Источниками и получателями могут быть терминалы пользователей, компьютеры, принтеры или любые другие устройства передачи и/или обработки данных. При таком способе передачи одни и те же средства передачи информации разделяются между пакетами многих пользователей.

В сетях другого типа применяется техника коммутации каналов (цепей). Это широко распространённые и привычные нам телефонные сети. В таких сетях устанавливается отдельный путь передачи, который удерживается столько времени, сколько требуется для передачи. В настоящее время развертываются интегральные сети, объединяющие в себе как технику коммутации пакетов, так и технику коммутации каналов. В общем случае для функционирования сетей ЭВМ необходимо решить две проблемы: передать данные по назначению в правильном виде и своевременно;

поступившие по назначению данные пользователю должны быть распознаваемы и иметь надлежащую форму для их правильного использования.

Первая проблема связана с задачами маршрутизации и обеспечивается сетевыми протоколами (протоколами низкого уровня). Вторая проблема вызвана использованием в сетях разных типов ЭВМ, с разными кодами и синтаксисом языка. Эта часть проблемы решается путем введения протоколов высокого уровня. Таким образом, полная архитектура, ориентированная на конечного пользователя, включает в себя оба протокола. В качестве примера на рис. 1.1 приведена схема связи между пользователями А и В. К промежуточному узлу связи могут быть подключены конечные пользователи, и его задачей протоколов является предоставление конечным пользователям соответствующих услуг. В свою очередь, это две группы протоколов: протоколы, предоставляющие сетевые услуги, и протоколы высокого уровня обычно подразделяются дальше на отдельные уровни. Каждый уровень используется для предоставления определенной услуги в смысле только что перечисленных задач: правильная и своевременная доставка данных в распознаваемой форме.

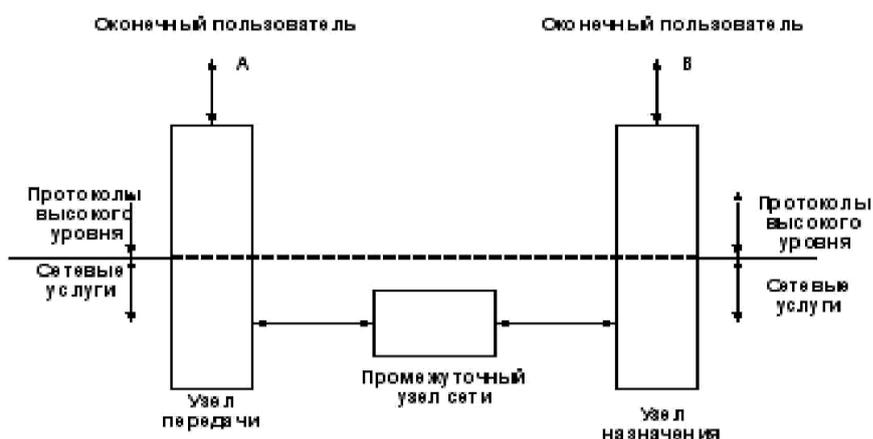


Рис. 1.1 Схема связи.

Разработанная эталонная модель взаимодействия открытых систем (ВОС) поддерживает концепцию, при которой каждый уровень предоставляет услуги вышестоящему уровню и базируется на основе нижележащего уровня и использует его услуги. Каждый уровень выполняет определенную функцию по передачи данных. Хотя они должны работать в строгой очередности, но каждый из уровней допускает несколько вариантов. Рассмотрим эталонную модель. Она состоит из 7 уровней и имеет вид, представленный на рис. 1.2.

Большинство производителей стараются придерживаться модели OSI, но до сих пор пока нет изделий полностью ей удовлетворяющих. Большинство производителей применяют 3 или 4 уровня протоколов. Взаимосвязь уровней друг с другом осуществляются хорошо определенными интерфейсами.

Выбор 7 уровней был продиктован обычными соображениями инженерного компромисса, требующего одновременно создать семейство надёжных протоколов и приемлемой стоимости. При этом требовалось, во-первых, иметь достаточно количество уровней, чтобы каждый из них был не слишком сложный с точки зрения разработки подробных протоколов с правильными и выполнимыми спецификациями, и во-вторых, желательно иметь не много уровней, чтобы их интеграция и описание не стали слишком сложными.

Эталонная модель как раз и представляет из себя многоуровневую архитектуру, которая описывается стандартными протоколами и процедурами. Три нижних уровня предоставляют сетевые услуги. Протоколы, реализующие эти уровни, должны быть предусмотрены в каждом узле сети. Четыре верхних уровня предоставляют услуги самим конечным пользователям и таким образом, связаны с ними, а не с сетью.

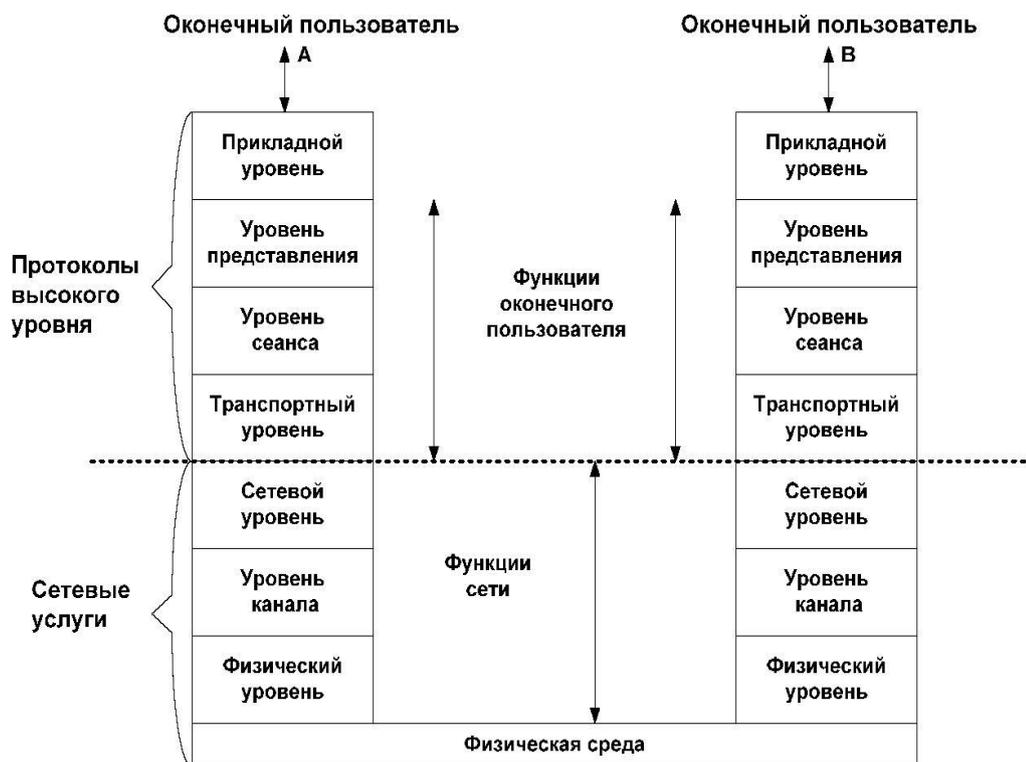


Рис. 1.2 Эталонная модель OSI

### *Физический уровень*

В этой части модели определяются физические, механические и электрические характеристики линий связи, составляющих ЛВС (кабелей, разъемов, оптоволоконных линий и т.п.). Можно считать, что этот уровень отвечает за аппаратное обеспечение. Хотя функции других уровней могут быть реализованы в соответствующих микросхемах, но все же они относятся к ПО. Функции физического уровня заключаются в гарантии того, что символы, поступающие в физическую среду передачи на одном конце канала, достигнут другого конца. При использовании этой нижестоящей услуги по транспортировке символов задача протокола канала состоит в обеспечении надежной (безошибочной) передаче блоков данных по каналу.

Такие блоки часто называют циклами, или кадрами. Процедура обычно требует: синхронизации по первому символу в кадре, распознавания конца кадра, обнаружения ошибочных символов, если таковые возникнут, и исправления таких символов каким-либо способом (обычно это делается

путем запроса на повторную передачу кадра, в котором обнаружены один или несколько ошибочных символов).

### *Уровень канала*

Уровень канала передачи данных и находящийся под ним физический уровень обеспечивают канал безошибочной передачи между двумя узлами в сети. На этом уровне определяются правила использования физического уровня узлами сети. Электрическое представление данных в ЛВС (биты данных, методы кодирования данных и маркеры) распознаются на этом и только на этом уровне. Здесь обнаруживаются (распознаются) и исправляются ошибки путем требований повторной передачи данных. Ввиду своей сложности, канальный уровень подразделяется на 2 подуровня MAC и LLC. Подуровень MAC (Media Access Control) связан с доступом к сети (передача маркера или обнаружение коллизий или столкновений) и ее управлением. Подуровень LLC находится выше уровня MAC и связан с передачей и приемом использованных сообщений.

### *Сетевой уровень*

Функция сетевого уровня состоит в том, чтобы установить маршрут для передачи данных по сети или при необходимости через несколько сетей от узла передачи до узла назначения. Этот уровень предусматривает также управление потоком или перегрузками с целью предотвращения переполнения сетевых ресурсов (накопителей в узлах и каналов передачи), которое может привести к прекращению работы. При выполнении этих функций на сетевом уровне используется услуга нижестоящего уровня – канала передачи данных, обеспечивающего безошибочное поступление по сетевому маршруту блока данных, введенного в канал на противоположном конце. В сети с коммутацией пакетов блоками данных, передаваемых по сетевому маршруту от одного конца к другому, как говорилось выше, являются пакеты. Блоки или кадры данных, передаваемые по каналу связи через сеть, состоят из пакетов плюс управляющей информации в виде заголовков и окончаний, добавляемых к пакету непосредственно перед его

отправлением из узла. Эта управляющая информация дает возможность принимающему узлу на другом конце канала выполнить требуемую синхронизацию и обнаружение ошибок. В каждом принимающем узле управляющая информация отделяется от остальной части пакета, а затем вновь добавляется, когда этот узел в свою очередь передает пакет по каналу в следующий соседний узел. Описанный принцип добавления управляющей информации к данным в архитектуре ВОС расширен и в нём включена возможность добавления управляющей информации на каждом уровне архитектуры (рис. 1.3 многоуровневую архитектуру) может быть легко изменена. В результате получается прозрачность для вышестоящего уровня при условии, что сигналы сопряжения, проходящие между уровнями, поддерживаются неизменными (оконечные пользователи могут ощущать изменение характеристик вследствие того, что характеристики передачи, задержек и блокировок могут зависеть от конкретной реализации архитектуры).

Таким образом, пакетная передача через сеть от одного конечного пользователя к другому в общем случае состоит в передаче фактической (полезной) информации, плюс управляющей информации, добавляемой на различных уровнях и подлежащей удалению, когда пакет поступает по назначению и начинает восстанавливаться на своем пути через эти уровни.

Оконечному пользователю сеть представляется как “прозрачный трубопровод”, основная задача которого, - передать по маршруту блоки данных от источника к получателю, доставив их своевременно в желаемый конец. Тогда задача верхних уровней – фактическая доставка данных в правильном виде и распознаваемой форме. Эти верхние уровни не знают о существовании сети. Они обеспечивают только требующуюся от них услугу.

#### *Транспортный уровень*

Нижний из верхних уровней ВОС, транспортный уровень, обеспечивает надежный, последовательный обмен данными между двумя конечными

пользователями. Для этой цели на транспортном уровне используется услуга сетевого уровня.

Он управляет также потоком, чтобы гарантировать правильный прием блоков данных. Вследствие различия конечных устройств, данные в системе могут передаваться с разными скоростями, поэтому, если не действует управление потоками, более медленные системы могут быть переполнены быстродействующими. Когда в процессе обработки находится больше одного пакета, транспортный уровень контролирует очередность прохождения компонент сообщения. Если приходит дубликат принятого ранее сообщения, то данный уровень опознает это и игнорирует сообщение.

### Уровень сеанса

Функции этого уровня состоят в координации связи между двумя прикладными программами, работающими на разных рабочих станциях. Он также предоставляет услуги вышестоящему уровню представления. Это происходит в виде хорошо структурированного диалога. В число этих функций входит создание сеанса, управление передачей и приемом пакетов сообщений в течении сеанса и завершение сеанса. Этот уровень при необходимости также управляет переговорами, чтобы гарантировать правильный обмен данными.

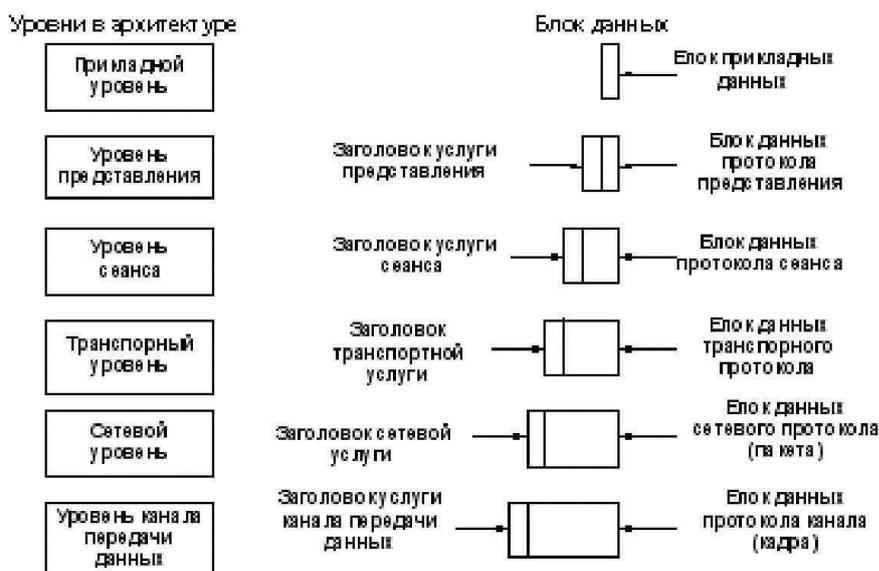


Рис. 1.3 Многоуровневая архитектура

В случае уровня канала может быть добавлено также окончание, но принцип последовательного добавления к пакету данных управляющей информации сохраняется. На каждом уровне блок данных принимается от вышестоящего уровня, к данным добавляется управляющая информация, и блок данных передается нижестоящему уровню. На приемном конце соответствующего уровня архитектуры используется только заголовок (управляющая информация). При этом подходе данный уровень не “просматривает” блок данных, который он получает от вышестоящего уровня, следовательно, уровни самостоятельны и изолированы друг от друга. Указанная особенность приводит к ценному свойству концепции многоуровневой архитектуры – уровни могут удаляться и заменяться новыми. Реализация (программные продукты, воплощающие пользователем сеансовой услуги (т.е. сторонами уровня представления и вышестоящим уровнем) может состоять из нормального или ускоренного обмена данными. Он может быть дуплексным, т.е. одновременной двусторонней передачей, когда каждая сторона имеет возможность независимо вести передачу, или полудуплексной, т.е. с одновременной передачей только в одну сторону. В последнем случае для передачи управления с одной стороны к другой применяются специальные метки. Уровень сеанса предоставляет услугу синхронизации для преодоления любых обнаруженных ошибок. При этой услуге метки синхронизации должны вставляться в поток данных пользователями услуги сеанса. Если будет обнаружена ошибка, то сеансовое соединение должно быть возвращено в определенное состояние, пользователи должны вернуться в установленную точку диалогового потока, сбросить часть переданных данных и затем восстановит передачу, начиная с этой точки.

Уровень сеанса предусматривает также при желании функцию управления активностью. При осуществлении этой функции диалог может быть разбит на отрезки активности, каждый из которых может быть прерван и продолжен в любой момент, начиная со следующего отрезка активности.

### *Уровень представления*

Наконец, уровень, представления управляет и преобразует синтаксис блоков данных, которыми обмениваются оконечные пользователи. Такая ситуация может возникать в неоднотипных ПК (IBM PC, Macintosh, DEC, Next, Burrough), которым необходимо обмениваться данными. Назначение - преобразование синтаксических блоков данных.

### *Прикладной уровень*

Протоколы прикладного уровня придают соответствующую семантику или смысл обмениваемой информации. Этот уровень является пограничным между ПП и процессами модели OSI. Сообщение, предназначенное для передачи через компьютерную сеть, попадает в модель OSI в данной точке, проходит через уровень 1 (физический), пересылается на другой PC, и проходит от уровня 1 в обратном порядке до достижения ПП на другом PC через ее прикладной уровень. Таким образом, прикладной уровень обеспечивает взаимопонимание двух прикладных программ на разных компьютерах.

## **1.3 Разновидности каналов связи**

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. Среды 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. Примерами линий передачи данных являются коаксиальный кабель, витая пара проводов, световод.

Характеристиками линий передачи данных являются зависимости затухания сигнала от частоты и расстояния. Затухание принято оценивать в

децибеллах,  $1 \text{ дБ} = 10 \cdot \lg(P1/P2)$ , где  $P1$  и  $P2$  - 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) передается  $N$  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)  $V = 2 \cdot F \cdot \log_2 k$  бит/с, так как  $V = \log_2 k / t$ , где  $t$  - длительность переходных процессов, приблизительно равная  $3T_B$  а  $T_B = 1 / (2 \cdot p - F)$ , здесь  $k / 1 + A$ ,  $A$  - 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 передается 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 (ПД) 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.

Первые сети ПД были аналоговыми, поскольку использовали распространенные телефонные технологии. Но в дальнейшем устойчиво растет доля цифровых коммуникаций (это каналы типа E1/T1, ISDN, сети Frame Relay, выделенные цифровые линии и др.).

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 ПД различают одно- и многоканальные средства ПД. В локальных вычислительных сетях и в цифровых каналах передачи данных обычно используют временное мультиплексирование, в аналоговых каналах - частотное разделение.

Если канал ПД монопольно используется одной организацией, то такой канал называют выделенным, в противном случае канал является разделяемым или виртуальным (общего пользования). К передаче информации имеют прямое

отношение телефонные сети, вычислительные сети передачи данных, спутниковые системы связи, системы сотовой радиосвязи.

## **ВЫВОДЫ**

В данной главе рассмотрены характеристики и классификации информационных сетей. Подробно изложена многоуровневая архитектура информационных сетей. Эталонная модель представляет из себя многоуровневую архитектуру, которая описывается стандартными протоколами и процедурами. Три нижних уровня предоставляют сетевые услуги. Протоколы, реализующие эти уровни, должны быть предусмотрены в каждом узле сети. Четыре верхних уровня предоставляют услуги самим конечным пользователям и таким образом, связаны с ними, а не с сетью.

Уровень канала передачи данных и находящийся под ним физический уровень обеспечивают канал безошибочной передачи между двумя узлами в сети. На этом уровне определяются правила использования физического уровня узлами сети. Электрическое представление данных в ЛВС (биты данных, методы кодирования данных и маркеры) распознаются на этом и только на этом уровне. Здесь обнаруживаются (распознаются) и исправляются ошибки путем требований повторной передачи данных.

Таким образом, пакетная передача через сеть от одного конечного пользователя к другому в общем случае состоит в передаче фактической информации, плюс управляющей информации, добавляемой на различных уровнях и подлежащей удалению, когда пакет поступает по назначению и начинает восстанавливаться на своем пути через эти уровни.

## 2. ОБЗОР ПРОТОКОЛА HDLC

### 2.1 Понятие протокола канального уровня

Для создания надежного механизма передачи данных между двумя станциями необходимо определить протокол, который позволит принимать и передавать различные данные по каналам связи. Протоколы представляют собой просто набор условий (правил), которые регламентируют формат и процедуры обмена информацией между двумя или несколькими независимыми устройствами или процессами. Протокол имеет три важнейших элемента: синтаксис, семантику и синхронизацию. Синтаксис протокола определяет поля; например, может быть 16-байтовое поле для адресов, 32-байтовое поле для контрольных сумм и 512 байт на пакет. Семантика протокола придает этим полям значение: например, если адресное поле состоит из всех адресов, это «широковещательный» пакет. Синхронизация – количество битов в секунду – это скорость передачи данных. Она важна не только на самых низких уровнях протокола, но и на высших.

Протокол канального обеспечивает следующие функции:

- управление передачей данных через физический канал организованный на первом уровне;
- проверка информационного канала;
- формирование кадра т.е. окаймление передаваемых данных служебными символами данного уровня;
- контроль данных;
- обеспечение прозрачности информационного канала;
- управление каналом передачи данных;

Данный протокол занимает второй уровень в многоуровневой организации управления сетью.

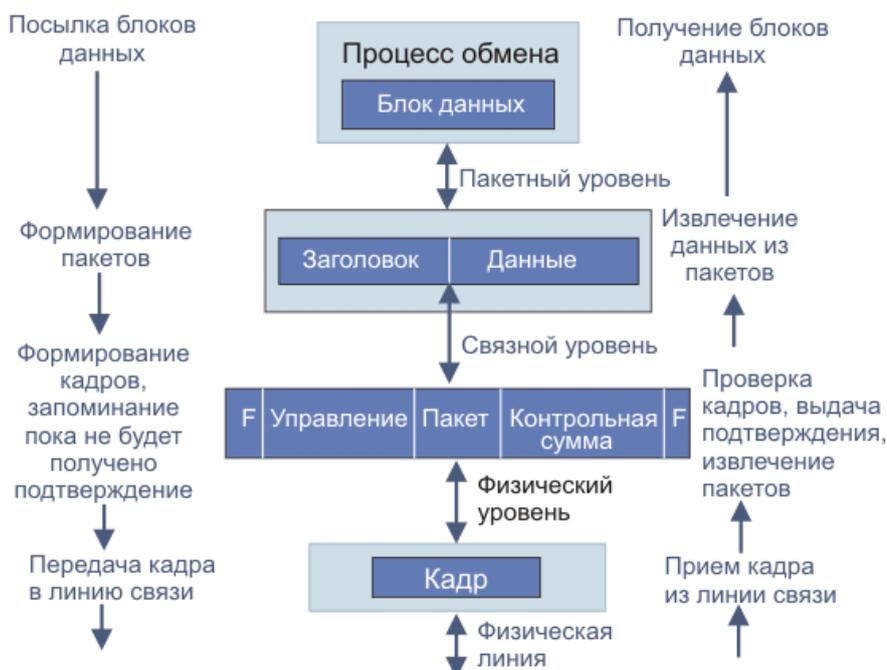


Рисунок 2.1 Процесс обмена информации при применении протокола HDLC

## 2.2 Типы, логические состояния и режимы работы станций

Существует *три типа станций HDLC*:

*Первичная станция* (ведущая) управляет звеном передачи данных (каналом). Несет ответственность за организацию потоков передаваемых данных и восстановление работоспособности звена передачи данных. Эта станция передает кадры команд вторичным станциям, подключенным к каналу. В свою очередь она получает кадры ответа от этих станций. Если канал является многоточечным, главная станция отвечает за поддержку отдельного сеанса связи с каждой станцией, подключенной к каналу.

*Вторичная станция* (ведомая) работает как зависимая по отношению к первичной станции (ведущей). Она реагирует на команды, получаемые от первичной станции, в виде ответов. Поддерживает только один сеанс, а

именно только с первичной станцией. Вторичная станция не отвечает за управление каналом.

*Комбинированная станция* сочетает в себе одновременно функции первичной и вторичной станции. Передает как команды, так и ответы и получает команды и ответы от другой комбинированной станции, с которой поддерживает сеанс.

*Три логических состояния*, в которых могут находиться станции в процессе взаимодействия друг с другом.

*Состояние логического разъединения (LDS)*. В этом состоянии станция не может вести передачу или принимать информацию. Если вторичная станция находится в нормальном режиме разъединения (NDM – Normal Disconnection Mode), она может принять кадр только после получения явного разрешения на это от первичной станции. Если станция находится в асинхронном режиме разъединения (ADM – Asynchronous Disconnection Mode), вторичная станция может инициировать передачу без получения на это явного разрешения, но кадр должен быть единственным кадром, который указывает статус первичной станции. Условиями перехода в состояние LDS могут быть начальное или повторное (после кратковременного отключения) включение источника питания; ручное управление установлением в исходное состояние логических цепей различных устройств станции и определяется на основе принятых системных соглашений.

*Состояние инициализации (IS)*. Это состояние используется для передачи управления на удаленную вторичную / комбинированную станцию, ее коррекции в случае необходимости, а также для обмена параметрами между удаленными станциями в звене передачи данных, используемыми в состоянии передачи информации.

*Состояние передачи информации (ITS)*. Вторичной, первичной и комбинированным станциям разрешается вести передачу и принимать информацию пользователя. В этом состоянии станция может находиться в режимах NRM, ARM и ABM, которые описаны ниже.

*Три режима работы* станции в состоянии передачи информации, которые могут устанавливаться и отменяться в любой момент.

*Режим нормального ответа (NRM – Normal Response Mode)* требует, чтобы прежде, чем начать передачу, вторичная станция получила явное разрешение от первичной. После получения разрешения вторичная станция начинает передачу ответа, который может содержать данные. Пока канал используется вторичной станцией, может передаваться один или более кадров. После последнего кадра вторичная станция должна снова ждать явного разрешения, прежде чем снова начать передачу. Как правило, этот режим используется вторичными станциями в многоточечных конфигурациях звена передачи данных.

*Режим асинхронного ответа (ARM – Asynchronous Response Mode)* позволяет вторичной станции инициировать передачу без получения явного разрешения от первичной станции (обычно, когда канал свободен, – в состоянии покоя). Этот режим придает большую гибкость работы вторичной станции. Могут передаваться один или несколько кадров данных или управляющая информация, отражающая изменение статуса вторичной станции. ARM может уменьшить накладные расходы, поскольку вторичная станция, чтобы передать данные, не нуждается в последовательности опроса. Как правило, такой режим используется для управления соединенными в кольцо станциями или же в многоточечных соединениях с опросом по цепочке. В обоих случаях вторичная станция может получить разрешение от другой вторичной станции и в ответ на него начать передачу. Таким образом разрешение на работу продвигается по кольцу или вдоль соединения.

*Асинхронный сбалансированный режим (ABM – Asynchronous Balanse Mode)* используют комбинированные станции. Комбинированная станция может инициировать передачу без получения предварительного разрешения от другой комбинированной станции. Этот режим обеспечивает двусторонний обмен потоками данных между станциями и является основным (рабочим) и наиболее часто используемым на практике

*Три способа конфигурирования канала* для обеспечения совместимости взаимодействий между станциями, использующих основные элементы процедур HDLC и способных в процессе работы менять свой статус (первичная, вторичная, комбинированная):

*Несбалансированная конфигурация (UN – Unbalanced Normal)* обеспечивает работу одной первичной станции и одной или большего числа вторичных станций в конфигурации односточечной или многоточечной, полудуплексной или полнодуплексной, с коммутируемым каналом и с некоммутируемым. Конфигурация называется несбалансированной потому, что первичная станция отвечает за управление каждой вторичной станцией и за выполнение команд установления режима.

*Симметричная конфигурация (UA – Unbalanced Asynchronous)* была в исходной версии стандарта HDLC и использовалась в первых сетях. Эта конфигурация обеспечивает функционирование двух независимых двухточечных несбалансированных конфигураций станций. Каждая станция обладает статусом первичной и вторичной, и, следовательно, каждая станция логически рассматривается как две станции: первичная и вторичная. Главная станция передает команды вторичной станции на другом конце канала, и наоборот. Несмотря на то, что станция может работать как в качестве первичной, так и вторичной станции, которые являются самостоятельными логическими объектами, реальные команды и ответы мультиплексируются в один физический канал. Этот подход в настоящее время используется редко.

*Сбалансированная конфигурация (BA – Balanced Asynchronous)* состоит из двух комбинированных станций, метод передачи – полудуплексный или дуплексный, канал – коммутируемый или некоммутируемый. Комбинированные станции имеют равный статус в канале и могут несанкционированно посылать друг другу трафик. Каждая станция несет одинаковую ответственность за управление каналом.

## 2.3 Управление потоком

Управление потоком в HDLC осуществляется с помощью передающих и принимающих окон. Окно устанавливается на каждом конце канала связи, чтобы обеспечить резервирование ресурсов обеих станций. Этими ресурсами могут быть ресурсы вычислителя или пространство буфера. В большинстве случаев окно обеспечивает и буферное пространство, и правила нумерации (сообщений). Окно устанавливается во время инициирования сеанса связи между станциями. Если станция А и станция В должны обменяться данными, А резервирует окно для В, а В резервирует окно для А.

Использование окон необходимо для полнодуплексных протоколов, потому что они подразумевают непрерывный поток кадров в принимающий узел без периодических подтверждений с остановкой и ожиданием.

Переменные состояния станции  $V(S)$  и  $V(R)$ . Окна в принимающем и передающем узлах управляются переменными состояниями, которые представляют по сути состояние счетчика. Передающий узел поддерживает переменную состояния посылки  $V(S)$ . Это порядковый номер следующего по очереди I-кадра, который должен быть передан.

Принимающий узел поддерживает переменную состояния приема  $V(R)$ , которая содержит номер, который, как ожидается, является порядковым номером следующего I-кадра.  $V(S)$  увеличивается на 1 при передаче каждого кадра и помещается в поле порядкового номера посылки кадра. Получив кадр, принимающий узел производит проверку наличия ошибок передачи и сравнивает порядковый номер со своим  $V(R)$ .

Если кадр может быть принят, узел увеличивает  $V(R)$  на 1, помещает его в поле порядкового номера приема кадра подтверждения АСК и посылает этот кадр в узел-отправитель, завершая квитирование передачи. Если  $V(R)$  не равен порядковому номеру посылки в кадре или обнаружена ошибка, значит, что-то произошло, и после тайм-аута в узел-отправитель посылается NAK [с

порядковым номером приема, содержащим значение  $V(R)$ ]. В большинстве протоколов этот NAK называется Неприем (REJ) или Выборочный неприем (SREJ). Значение  $V(R)$  уведомляет передающее устройство ООД о том, что ожидается посылка нового кадра. Т. к. передатчик восстанавливает старое значение  $V(S)$  и повторяет передачу кадра, порядковый номер которого совпадает со значением  $V(S)$ .

Во многих системах для  $V(S)$  и  $V(R)$  у порядковых номеров в кадре используются числа 0-7. Если переменные состояния в результате последовательного увеличения достигли 7, то, начиная с 0, эти числа снова используются. Вследствие повторного использования чисел устройствам станциям не разрешено посылать кадр с порядковым номером, который не был подтвержден. Например, протокол должен дождаться подтверждения кадра с номером 6, прежде чем он опять использует  $V(S)=6$ . Этот процесс показан на рис.2.1. Здесь кадры с 6 по 4 еще не подтверждены. Если бы был послан еще один кадр с порядковым номером 6, соответствующее подтверждение АСК с номером 6 не позволило бы определить, приход какого кадра с порядковым номером 6 подтверждается. Использование номеров 0-7 позволяет семи кадрам быть в активном состоянии, прежде, чем "закроется" окно. Несмотря на то что диапазон 0-7 дает восемь порядковых номеров,  $V(R)$  содержит значение следующего ожидаемого кадра, что ограничивает число активных кадров до 7.

## 2.4 Формат кадра HDLC

На канальном уровне используется термин кадр для обозначения независимого объекта данных, передаваемого от одной станции к другой (рис. 2.1).

*Флаг.* Все кадры должны начинаться и заканчиваться полями флага «01111110». Станции, подключенные к каналу, постоянно контролируют

двоичную последовательность флага. Флаги могут постоянно передаваться по каналу между кадрами HDLC. Для индексации исключительной ситуации в канале могут быть посланы семь подряд идущих единиц. Пятнадцать или большее число единиц поддерживают канал в состоянии покоя. Если принимающая станция обнаружит последовательность битов не являющихся флагом, она тем самым уведомляется о начале кадра, об исключительной (с аварийным завершением) ситуации или ситуации покоя канала. При обнаружении следующей флаговой последовательности станция будет знать, что поступил полный кадр.

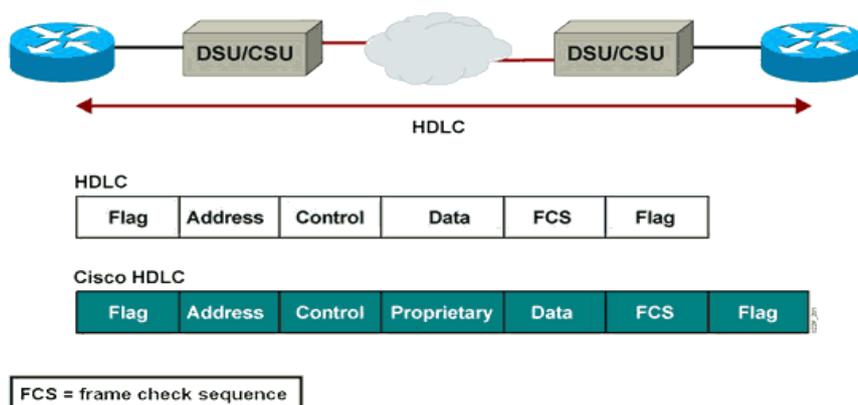


Рисунок 2.1 Формат кадра HDLC

Формат кадра HDLC								
Флаг	Адрес	Управляющее поле			Информационное поле		CRC	Флаг
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	

Рис. 2.1. Формат кадра и управляющего поля HDLC, где:

$N(S)$  – 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. (табл.2.1)

Таблица 2.1 Правила адресации

Правила адресации			
Первичная станция А	– Команда (Адрес В) →	Вторичная станция В	Несбалансир. конфигурация
	< – Ответ (Адрес В) –		
Комбинир. станция А	– Команда (Адрес В) →	Комбинир. станция В	Сбалансир. конфигурация
	< – Ответ (Адрес В) –		
	< – Команда (Адрес А) –		
	– Ответ (Адрес А) →		

*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 (рис. 1) определяют кадры трех типов: информационные (I), супервизорные (S) и нумерованные (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.

*Информационное поле* содержит действительные данные пользователя. Информационное поле имеется только в кадре информационного формата. Его нет в кадре супервизорного или ненумерованного формата. [Примечание: кадры «UI – ненумерованная информация» и «FRMR – Неприем кадра» ненумерованного формата имеют информационное поле].

*Поле CRC* (контрольная последовательность кадра) используется для обнаружения ошибок передачи между двумя станциями. Передающая станция осуществляет вычисления над потоком данных пользователя, и результат этого вычисления включается в кадр в качестве поля CRC. В свою очередь, принимающая станция производит аналогичные вычисления и сравнивает полученный результат с полем CRC. Если имеет место совпадение, велика вероятность того, что передача произошла без ошибок. В случае несовпадения, возможно, имела место ошибка передачи, и принимающая станция посылает отрицательное подтверждение, означающее, что необходимо повторить передачу кадра. Вычисление CRC называется циклическим контролем по избыточности и использует некоторый производящий полином в соответствии с рекомендацией МККТТ V.41. Этот метод позволяет обнаруживать всевозможные кортежи ошибок длиной не более 16 разрядов, вызываемые одиночной ошибкой, а также 99,9984% всевозможных более длинных кортежей ошибок.

## **2.5 Управляющее поле HDLC**

Управляющее поле (рис. 2.1) определяет тип кадра и используется для реализации механизма управления потоком между передающей и принимающей станциями. На табл. 2.2 представлены команды и ответы, используемые в случае сбалансированной и несбалансированной конфигураций канала. Отметим, что в каждом верхнем прямоугольнике содержатся три команды: SNRM, SARM, SABM.

Таблица 2.2 Команды и ответы

Несбалансированный (UN)		Несбалансированный (UA)		Сбалансированный (UB)	
Первичная	Вторичная	Первичная	Вторичная	Первичная	Вторичная
Команда	Ответ	Команда	Ответ	Команда	Ответ
I	I	I	I	I	I
RR	RR	RR	RR	RR	RR
RNR	RNR	RNR	RNR	RNR	RNR
SNRM	UA	SARM	UA	SABM	UA
DISC	DM	DISC	DM	DISC	DM
	FRMR		FRMR		FRMR

Эти команды являются командами установки режима. HDLC требует, чтобы в одном из трех режимов была установлена сбалансированная или несбалансированная конфигурация. Действительный формат управляющего поля (информационный, супервизорный или нумерованный) определяет то, как это поле кодируется или используется. Самым простым форматом является информационный формат. Содержимое управляющего поля для этого формата показано на рис. 2.1. Управляющее поле информационного кадра содержит два порядковых номера. Номер N(S) (Порядковый номер отправки) связан с порядковым номером передаваемого кадра. N(R) (Порядковый номер приема) означает порядковый номер следующего кадра,

который ожидается принимающей станцией.  $N(R)$  выступает в качестве подтверждения предыдущих кадров. Например, если поле  $N(R)$  установлено в 4, станция, получив  $N(R)=4$ , знает, что передача кадров 0, 1, 2 и 3 завершилась успешно и что станция, с которой производится обмен данными, ожидает, что следующий кадр будет иметь порядковый номер посылки  $N(S)=4$ . Поле  $N(R)$  обеспечивает включающее подтверждение (квитирование), то есть  $N(R)=4$  включает подтверждение не только одного предшествующего сообщения. Переменные состояния посылки  $V(S)$  и состояния приема  $V(R)$ , рассмотренные нами ранее, используются для формирования полей  $N(S)$  и  $N(R)$  протокола HDLC.

Пятый двоичный разряд, бит P/F или бит опроса / окончания принимается во внимание только тогда, когда он установлен в 1. Бит P/F называется битом P, когда он используется первичной станцией, и битом F, когда он используется вторичной станцией. Он используется первичной и вторичной станциями для выполнения следующих функций:

Первичная станция использует бит P для санкционирования передачи кадра статуса от вторичной станции. P также может означать опрос.

Вторичная станция отвечает на бит P кадром данных или состояния с битом F. Бит F может также означать окончание передачи вторичной станцией в режиме нормального ответа (NRM).

Только один бит P (ожидающий ответа в виде F бита) может быть активным в канале в любой момент времени. Если некоторый бит P установлен в 1, он может быть использован в качестве контрольной точки. То есть  $P=1$  как бы говорит: ответьте мне, потому что я хочу знать ваш статус. Контрольные точки играют большую роль в реализации механизма управления трафиком. Это также способ устранения неопределенностей и отмены накопленных транзакции. Бит P/F может использоваться и интерпретироваться следующим образом:

В режиме NRM вторичная станция не может вести передачу, пока не будет получена команда с установленным в 1 битом P. Первичная станция

может запросить информационные (I) кадры путем посылки кадра с установленным в 1 битом P или путем посылки некоторых супервизорных (S) кадров (RR, REJ или SREJ) с установленным в 1 битом P.

В режимах ARM и ABM информационные кадры могут передаваться без получения полномочий на передачу с помощью команды, имеющей бит P. Поэтому бит P используется в этом случае для запроса ответа с установленным в 1 битом F так быстро, насколько это возможно. Например, в случае двунаправленной одновременной (полнодуплексной) передачи, когда по получении команды с установленным в 1 битом P передачу ведет вторичная станция, бит F устанавливается в 1 в самом первом очередном ответе. Передача кадра с установленным в 1 битом F не требует, чтобы вторичная станция прекратила передачу. Вслед за кадром с установленным в 1 битом F могут быть еще переданы кадры. В режимах ARM и ABM не следует интерпретировать бит F как окончание передачи вторичной станцией; его следует просто считать индикатором ответа на предыдущий кадр.

## **2.6 Процессы передачи в протоколе HDLC.**

На рисунках 2.2,2.3,2.4,2.5,2.6 показаны различные виды коммуникации:

- асинхронный сбалансированный режим с полудуплексным потоком данных (рис. 2.2);
- асинхронный сбалансированный режим с полнодуплексным потоком данных (рис. 2.3);
- восстановление после ошибок (исправление ошибок) по методу Возвращение-на-N (Go-Back-N) (контрольная точка) (рис.2.4);
- восстановление после ошибок по методу Возвращение-на-N (Отвергнуть) (рис. 2.5);

- восстановление после ошибок по методу Выборочного неприема (Selective Reject) (рис.2.6);
- SDLC с многоточечной полнодуплексной передачей (рис.2.7).

Условные обозначения, используемые на рисунках. Рисунки представляют собой как бы "логические снимки", сделанные в отдельные интервалы времени ( $t$ ,  $t+1$  и т.д.). Обозначения, находящиеся во временном "окне", отражают содержание кадра HDLC (или некоторого подмножества протокола, например, LAPB), передаваемого станциями А и В в конкретное время.

Весьма маловероятно, что две станции начнут передачу строго в один и тот же момент времени, но для упрощения объяснения мы будем придерживаться этого предположения. Например, временное окно станции А могло бы быть изображено более широким, чем окно станции В, что означало бы, что станцией А передается более длинный кадр, но неравные окна не оправдано усложнили бы и без того сложную тему. Если согласится с этой небольшой аномалией, принципы, которые поясняются на рисунках, остаются в силе. Кроме того, иллюстрации полнодуплексного метода показывают некоторые временные окна, относящиеся к каналу, который находится в состоянии покоя. Это может иметь место или нет в зависимости от того, как загружены станции.

Смысл обозначений такой:

А,В - Адрес станции в заголовке кадра.

I - Информационный кадр.

S=x - Порядковый номер посылки x.

R=x - Порядковый номер приема x.

RR,SNRM,SABM,REJ,SREJ - Команды и ответы.

P/F - Бит опроса/окончания установлен в 1.

Напомним, что порядковый номер приема  $N(R)$  означает включающее подтверждение всего переданного и принятого трафика. Номер в этом поле в действительности представляет собой величину, на 1 большую номера

последнего подтвержденного кадра. Например, R=4 означает, что подтверждены кадры 0,1,2 и 3 и что приемник ожидает, что следующий кадр будет иметь 4 в поле порядкового номера посылки передающей станции. По ходу описания процесса будем обсуждать бит P/F там, где это необходимо.

Все рисунки сопровождаются кратким описанием событий в каждый момент времени. Можно заметить, что в иллюстрациях в качестве адреса станции используется либо А, либо В. Как отмечалось ранее, правилами протокола HDLC определено, какой адрес (передающей или принимающей станции) помещается в поле адреса: команды используют адрес принимающего одноуровневого логического объекта уровня звена данных. Таким образом в случае станции с адресом А, если принятый кадр содержит А, это команда; если принятый кадр содержит В, это ответ. Описание событий для процесса, представленного на рис.2.2:

	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8
Ст. А	B, SABM P		B,I S=0 R=0	B,I,P S=1 R=0				A,RR F R=2	
Ст. В		B,UA F			B,RR F R=2	A,I S=0 R=2	A,I,P S=1 R=2		B,RR F R=2

Рис.2.2 Асинхронный сбалансированный режим с полудуплексным потоком данных (с использованием P/F для реализации "контрольной точки").

- t Станция А передает команду *Установить асинхронный сбалансированный режим (SABM)* с установленным битом P.
- t+1 Станция В отвечает *Ненумерованным подтверждением (UA)* с установленным битом F.

- t+2,3 Станция А посылает информационные кадры 0 и 1, устанавливает бит Р.
- t+4,5,6 Станция В подтверждает передачу станции А, посылая 2 в поле порядкового номера приема. Станция В, кроме того, передает информационные кадры 0 и 1.
- t+7 Станция А подтверждает кадры 0 и 1 станции В, порядковый номер 2 в поле приема.
- t+8 Станция В также подтверждает последний переданный А кадр с номером 2 и объявляет, что ему нечего передавать битом F. Отметим, что поле приема станции В сохраняло значение 2.

За исключением рис.2.8, соглашения относительно адресации, принятые в иллюстрациях, соответствуют подмножеству HDLC-LAPB (сбалансированной процедуре доступа к звену). Этот широко используемый протокол требует, чтобы все информационные (I) кадры были командными кадрами. Вследствие этого он содержит адрес приемника. Хотя все эти примеры недопустимы в LAPB, для наглядности иллюстраций используется некоторая непротиворечивая схема адресации.

Ниже приведены моменты времени и события для процесса, представленного на рис.2.3:

	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8
Станция А передает	B,I S=0 R=0	B,I,P S=1 R=1			B,I S=2 R=3	B,RR, P R=4		B,I S=3 R=5	B,RR P R=5
Станция В передает	A,I S=0 R=0	A,I S=1 R=1	B,RR F R=2	A,I S=2 R=2	A,I S=3 R=2	A,I S=4 R=3	B,RR F R=3	A,I S=5 R=3	A,RR P R=4

Рис.2.3 Асинхронный сбалансированный режим с полнодуплексным потоком данных (Р не останавливает потока данных).

Предполагается, что в предыдущих кадрах был установлен асинхронный сбалансированный режим:

- $t$  Обе станции, А и В передают информационный кадр с порядковым номером посылки 0.
- $t+1$  Станции А и В посылают подтверждение приема кадров с номером 0, используя порядковые номера приема, равные 1. Они также передают информационные кадры с порядковыми номерами посылки, равными 1. Станция А посылает разрешение на ответ, устанавливая в 1 бит Р.
- $t+2$  Станция В немедленно отвечает командой *Готов к приему* с битом F на полученную команду от Станции А с установленным Р-битом. Чтобы подтвердить кадр с номером 1 от станции А, В использует порядковый номер приема 2. В режиме АВМ Станция В может продолжать передачу в следующий момент времени.
- $t+3$  Станция В передает информационный кадр 2 и повторно подтверждает принятый кадр 1
- $t+4$  Станция А посылает информационный кадр 2 и подтверждает кадры 1 и 2 станции В, используя порядковый номер приема 3. Станция В посылает информационный кадр 3.
- $t+5$  Станции А посылать нечего, но она подтверждает кадр с номером 3 станции В, используя порядковый номер приема 4, и требует ответа, устанавливая в 1 бит Р. Станция В передает информационный кадр 4 и подтверждает кадр с номером 2 станции А, используя порядковый номер приема 3.
- $t+6$  Станция В отвечает на предыдущий бит Р установкой бита F в 1.

- t+7 Станция А посылает информационный кадр 3 и подтверждает кадр 4 станции В, используя порядковый номер приема 5. Станция В посылает информационный кадр 5.
- t+8 Ни у одной станции нет данных для передачи. Станция А посылает *Готов к приему* (RR), чтобы инициировать прием кадра 6. Станция В подтверждает кадр 3 станции А, используя порядковый номер приема 4.

Рис.2.4, 2.5, 2.6 являются примерами того, как в протоколе HDLC обрабатываются ошибки передачи. На рис.6 показано использование поля порядкового номера приема N(R) для отрицательного (NAK) подтверждения кадра. На рис.2.5 показано использование *Неприема* (REJ), а рис.2.6 иллюстрирует использование *Выборочного неприема* (SREJ). Здесь рассматривается момент t продолжающегося сеанса, когда станция А передает кадр с номером 6. Ниже приведены моменты времени и события для процесса, показанного на рис.2.4 (не поддерживаемого протоколом LAPB)

	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8
Станция А передает	B,I S=6 R=4	B,I S=7 R=4 (ошибка)	B,I S=0 R=4	B,I,P S=1 R=4		B,I S=7 R=4	B,I S=0 R=4	B,I,P S=1 R=4	
Станция В передает					B,RR,F R=7				RR,F R=2

Рис.2.4 Восстановление по методу Возвращение-на-N (контрольная точка).

Рисунок иллюстрирует продолжающийся сеанс; Р и F используются для реализации восстановления:

- $t, t+1, 2, 3$  Станция А посылает информационные кадры 6, 7, 0 и 1. Отметим, что, поскольку 7 является наибольшим допустимым порядковым номером, после 7 следует 0. Во время этого периода станция В обнаруживает ошибку в кадре 7. В  $t+3$  станция А посылает бит опроса, который производит такое же действие, как контрольная точка, т.е. разрешает ответ станции В.
- $t+4$  Станция В возвращает *Готов к приему*(RR) с новым номером посылки 7 и битом окончания F. Это означает, что станция В снова ожидает приема кадра 7 (и всех кадров, переданных после 7).
- $t+5, 6, 7$  Станция А повторно передает кадры 7, 0, и 1 и устанавливает бит Р в качестве контрольной точки.
- $t+8$  Станция В подтверждает кадры 7, 0 и 1 командой *Готов к приему* (RR) с порядковым номером приема 2 и установленным битом F.

Исключительное использование поля порядкового номера приема  $N(R)$  для отрицательного подтверждения кадра не рекомендуется для полнодуплексной передачи. Так как кадры передаются по каналу в обоих направлениях, порядковые номера посылки и приема часто перекрываются. Например, предположим, что кадр 4 станции А [ $N(S)=4$ ] передается примерно в то же время, что и кадр станции В, который содержит  $N(R)=4$ . Станция А может ошибочно заключить, что ее кадр 4 получен Станцией В с ошибкой, в то время как станция В просто указывает, что следующим она ожидает кадр 4. Более эффективный подход к исправлению ошибок состоит в явном указании ошибочного кадра. Рис.2.5 и 2.6 иллюстрируют два метода реализации явных отрицательных подтверждений NAK.

Ниже приведены моменты времени и события для процесса, который поясняется рис.2.5:

	t	t+1	t+2	t+3	t+4	t+5	t+6
--	---	-----	-----	-----	-----	-----	-----

Станция А передает	B,I S=6 R=4	B,I S=7 R=4(ошибка)	B,I S=0 R=4	B,I S=7 R=4	B,I S=0 R=4	B,I S=1 R=4	
Станция В передает			B,REJ F R=7				B,RR F R=2

Рис.2.5 Исправление ошибок с использованием метода Возвращение-на-N (REJ).

Рисунок иллюстрирует продолжающийся сеанс:

- $t, t+1, 2$  Станция А посылает информационные кадры 6, 7 и 0. Станция В обнаруживает ошибку в кадре 7 и немедленно посылает кадр *Неприем* с порядковым номером приема 7 и F-битом. Таким образом Станция В не ожидает санкции на реализацию контрольной точки и посылает ответ REJ (Неприем) с установленным F-битом. Если бы станция В посылала REJ в качестве команды (то есть с адресным полем, содержащимся в А), станция А должна была бы ответить кадрами RR, RNR или REJ. Однако, поскольку REJ - это ответ, станция А немедленно осуществит повторную передачу искаженного кадра.
- $t+3, 4, 5$  Станция А повторно передает кадры 7, 0 и 1 устанавливает бит Р в момент времени 5.
- $t+6$  Станция В подтверждает кадры 7, 0 и 1, используя *Готов к приему* и порядковый номер приема, равный 2. Отметим: для полnodуплексных систем бит Р/Ф обычно не используется для остановки потока данных, поскольку это снижает пропускную способность.

События для процесса, который представлен на рис.2.6 (не поддерживаемого протоколом LAPB):

	t	t+1	t+2	t+3	t+4	t+5
Станция А передает	B,I S=6 R=4	B,I S=7 R=4(ошибка)	B,I S=0 R=4	B,I S=7 R=4	B,I S=1 R=4	
Станция В передает			B,SREJ F R=7			B,RR F R=2

Рис.2.6. Исправление ошибок по методу Выборочный неприем (SREJ).

Рисунок иллюстрирует продолжающийся сеанс:

- t,t+1,2 Станция А передает информационные кадры 6,7 и 0. Станция В обнаруживает ошибку в кадре 7 и передает *Выборочный неприем* с порядковым номером 7. Станция В не требует RR, RNR или REJ, так как кадр в t+2 не является командой.
- t+3,4 Станция А передает повторно только кадр 7 и впервые передает кадр 1. Поскольку это *Выборочный неприем*, кадр 0 не передается повторно.
- t+5 Станция В подтверждает все принятые кадры ответом *Готов к приему* с порядковым номером приема 2.

## 2.7 Подмножества HDLC

Принятие базового множества (superset) протокола HDLC заложило прочную основу для реализации подмножества протокола HDLC. Некоторые подмножества используются в настоящее время в промышленности. Эти подмножества показаны на дереве классификаций сетей на рис.2.7.

Структура базового подмножества HDLC дает возможность бит-ориентированным протоколам распознавать и использовать в разнообразных приложениях одни и те же процедуры. Приложения требуют для выполнения своих функций различных режимов работы и различных подмножеств команд и ответов. Например, требования, выдвигаемые многоточечными интерактивными приложениями, отличаются от требований двухточечных неинтерактивных систем.

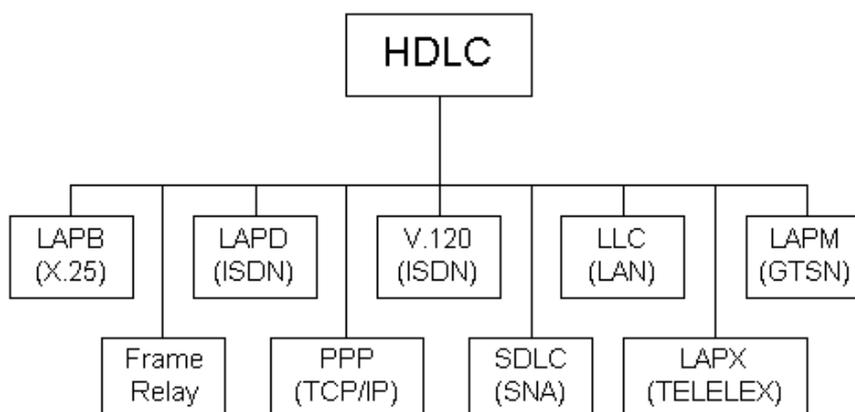


Рисунок 2.7. Семейство протокола HDLC.

Будем считать, что станция соответствует некоторому классу HDLC, если это оборудование реализует все команды и ответы, определенные для этого класса. Как указывалось ранее, тремя основными классами HDLC являются:

- класс UN (несбалансированный режим нормального ответа);
- класс UA (несбалансированный режим асинхронного ответа);
- класс BA (сбалансированный асинхронный режим).

HDLC предусматривает множество факультативных расширений основных классов. Эти расширения используются фирмами-поставщиками и пользователями HDLC для обеспечения большего разнообразия в структуре базового подмножества. Напомним, что опции и базовое множество HDLC показано на рис.2.7. Классы подмножеств обозначены сокращениями,

такими, как UN, UA или BA плюс факультативное расширение протокола HDLC, обозначаемые конкретным номером опции. Например, протокол, обозначенный BA-4, является сбалансированным асинхронным протоколом, предназначенный для передачи нумерованной информации (UI). Имея в виду эту классификационную схему, рассмотрим некоторые из основных подмножеств стандарта HDLC.

*LAP (Процедура доступа к звену)* является одним из наиболее ранних подмножеств HDLC. LAP основывается на команде SARM - Установить режим асинхронного ответа - для сбалансированной конфигурации. Реализация звена с LAB является несколько неуклюжей, так как требуется, чтобы прежде чем установлено звено, обе станции посылали SARM и UA. Она отличается от реализации широко используемой процедуры LAPB.

*LAPB (Сбалансированная процедура доступа к звену)* используется во всем мире несколькими частными вычислительными сетями и сетями общего пользования. LAPB -это некоторое подмножество репертуара команда/ответов HDLC. LAPB используется для поддержки широко распространенного протокола сети с пакетной передачей X.25. LAPB классифицируется как подмножество BA-2.5 HDLC. Это означает, что кроме использования асинхронного сбалансированного режима этот протокол использует также два функциональных решения: опции 2 и 8. Опция 2 делает возможным одновременный прием кадров в режиме двунаправленной передачи (рис.2.5). Опция 8 не допускает передачу полезной информации в кадрах ответа. Это не представляет какой-либо проблемы, так как в асинхронном сбалансированном режиме информация может представляться в командных кадрах, и поскольку обе физические станции являются логическими первичными станциями, обе могут представлять команды.

*LLC (Управление логическим звеном)* является стандартом, опубликованным Комитетом по стандартам IEEE 802 для локальных сетей. Стандарт допускает взаимодействие локальной сети с глобальной сетью. LLC использует подкласс базового множества HDLC; имеет классификационное

обозначение BA-2,4. Он использует сбалансированный асинхронный режим (BA) и функциональные расширения (опции 2 и 4).

LLC разработан так, чтобы его можно было поместить между уровнем локальной сети и уровнем глобальной сети. Подобная реализация показана на рис.2.8. Блок доступа к среде (MAU) содержит протоколы локальной сети, а LLC обеспечивает интерфейс для верхних уровней. Оба подуровня описаны более подробно в разделе локальных сетей. Все локальные сети, специфицированные IEEE 802, обеспечивают сервис без установления логического соединения (тип 1). Сервис, ориентированный на установление логического соединения (тип 2), может предоставляться факультативно.

<b>Сетевой уровень</b>	
<b>Канальный уровень</b>	Подуровень управления логическим звеном(LLC)
	Подуровень блока доступа к среде (MAU)
<b>Физический уровень</b>	

Рис.2.8. Управление логическим звеном (LLC)

В локальных сетях типа 1 также отсутствуют механизм управления потоком и восстановление данных после ошибок. Это обусловлено необходимостью снижения накладных расходов для высокоскоростных каналов (отсутствуют фазы установления и расторжения соединений, отсутствуют подтверждения приема). Целостность данных поддерживается в ЛВС верхними уровнями модели ВОС. Например, в TCP/IP сетях транспортным уровнем.

В локальных же сетях типа 2 эти функции (установление и расторжение соединений, механизм управления потоком посредством скользящего окна)

предусмотрены. Множества допустимых команд для LLC приведены на рис.2.8

*LAPD (Процедура доступа к D - каналу)* является еще одним подмножеством структуры HDLC, хотя ее расширение выходит за рамки HDLC. LABD предназначен для управления звеном данных в цифровых сетях с интеграцией служб ISDN, которая получает все большее развитие.

*LAPX (Расширенный LAPB)* это еще одно подмножества HDLC. Этот протокол (процедура) используется в терминальных системах и в развивающемся стандарте TELETEx. Это полудуплексный вариант HDLC.

	<b>Команды</b>	<b>Ответы</b>
Тип 1	UI - нумерованная информация	XID - идентификация станции
	XID - идентификация станции	XID - идентификация станции
	TEST- проверка	TEST- проверка
Тип 2 (I-формат) (S-формат)  (U-формат)	I - Информационный кадр	I - Информационный кадр
	RR - Готов к приему	RR - Готов к приему
	RNR - Не готов к приему	RNR - Не готов к приему
	REJ - Неприем	REJ - Неприем
	SABM - Установить ABM	UA - Ненумер.подтв., FRMR - Неприем
	DISC - разъединить	UA-Ненумер.подт., DM-Разъединение

Рис.2.9. Команды и ответы подуровня LLC локальной сети

*SDLC (Синхронное управление звеном данных)* является версией базового множества HDLC, разработанного компанией IBM. SDLC

использует несбалансированный режим нормального ответа и может быть классифицирован как UN-1,2,4,5,6,12. Обеспечивает поддержку двухточечных, многоточечных или кольцевых соединений. События для процесса, поддерживаемого протоколом SDLC, представлены на рис.2.10.

	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12	t+13
A	B,RR P		B SIM P		B SNRM P		C,RR P R=0	B,I S=0 R=0	B,I S=1 R=0		B,RR P R=0		C,RR P R=3	B,RR P R=2
B		B RIM F		B UA F		B UA F						B,I S=0 R=2	B,I,F S=1 R=2	
C								C,I S=0 R=0	C,I S=1 R=0	C,I,F S=2 R=0				

Рис.2.10. SDLC в полнодуплексной многоточечной системе передачи данных, где А - первичная станция, В - в режиме разъединения, С - в режиме нормального ответа

- t,t+1,2,3,4,5 Станция А сначала опрашивает статус (состояние) станции В. Станция В отвечает запросом режима инициализации (RIM). Станция А устанавливает В в режим инициализации (SIM), а затем в режим нормального ответа. В подтверждает оба режима.
- t+6 Станция А использует команду *Готов к приему (RR)* для опроса станции С путем установки бита Р.
- t+7,8 Станция А посылает информационные кадры 0 и 1 станции В. Одновременно станция С, отвечая на предыдущий опрос посылает

также информационные кадры 0 и 1 станции А по другому каналу полнодуплексной цепи.

- $t+9$  Станция С посылает информационный кадр 2 с установленным битом окончания F.
- $t+10$  Станция А опрашивает В для реализации контрольной точки (подтверждения).
- $t+11$  Станция В отвечает информационным кадром 0 и одновременно подтверждает принятые от А кадры 0 и 1, используя порядковый номер приема 2.
- $t+12$  Станция А подтверждает кадры 0, 1 и 2 станции С с помощью *Готов к приему (RR)* и порядкового номера приема 3. Станция В посылает информационный кадр 1 и устанавливает бит F в ответ на бит Р в  $t+1$ .
- $t+13$  Станция А подтверждает кадры 0 и 1 станции В с помощью команды *Готов к приему (RR)* и порядкового номера приема 2.
- $t+-$  Последующие события потребуют, чтобы станции А и В выдали ответы с битом F.

## ВЫВОДЫ

В данной главе приведён обзор протокола HDLC- управление потоком, формат кадра, управляющее поле и процессы передачи в данном протоколе. Рассмотрены также подмножества протокола HDLC. Структура базового подмножества HDLC дает возможность бит-ориентированным протоколам распознавать и использовать в разнообразных приложениях одни и те же процедуры. Приложения требуют для выполнения своих функций различных режимов работы и различных подмножеств команд и ответов.

## 3. ИММИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОТОКОЛА HDLC

### 3.1 Выбор языка программирования

Потребность в разработке и применении эффективных и адекватных реальной действительности компьютерных программ и технологий сегодня возрастает. Компьютерная технология незаменима, поскольку она даёт возможность оптимизировать и рационализировать управленческую функцию за счет применения новых средств сбора, передачи и преобразования информации.

Выпускная работа написана в программной среде Delphi. Delphi обладает широким набором возможностей, начиная от проектировщика форм и кончая поддержкой всех форматов популярных баз данных. Среда устраняет необходимость программировать такие компоненты Windows общего назначения, как метки, пиктограммы и даже диалоговые панели. Работая в Windows, вы неоднократно видели одинаковые «объекты» во многих разнообразных приложениях. Диалоговые панели (например, Choose File и Save File) являются примерами многократно используемых компонентов, встроенных непосредственно в Delphi, который позволяет приспособить эти компоненты к имеющейся задаче, чтобы они работали именно так, как требуется создаваемому приложению. Также здесь имеются предварительно определенные визуальные и не визуальные объекты, включая кнопки, объекты с данными, меню и уже построенные диалоговые панели. С помощью этих объектов можно, например, обеспечить ввод данных просто несколькими нажатиями кнопок мыши, не прибегая к программированию. Та часть, которая непосредственно связана с программированием интерфейса пользователя системой, получила название визуальное программирование.

Визуальное программирование как бы добавляет новое измерение при создании приложений, давая возможность изображать эти объекты на экране монитора до выполнения самой программы. Без визуального программирования процесс отображения требует написания фрагмента кода,

создающего и настраивающего объект «по месту». Увидеть закодированные объекты было возможно только в ходе исполнения программы. При таком подходе достижение того, чтобы объекты выглядели и вели себя заданным образом, становится утомительным процессом, который требует неоднократных исправлений программного кода с последующей прогонкой программы и наблюдения за тем, что в итоге получилось.

Благодаря средствам визуальной разработки можно работать с объектами, держа их перед глазами и получая результаты практически сразу. Способность видеть объекты такими, какими они появляются в ходе исполнения программы, снимает необходимость проведения множества операций вручную, что характерно для работы в среде, не обладающей визуальными средствами — вне зависимости от того, является она объектно-ориентированной или нет. После того, как объект помещен в форму среды визуального программирования, все его атрибуты сразу отображаются в виде кода, который соответствует объекту как единице, исполняемой в ходе работы программы.

Размещение объектов в Delphi связано с более тесными отношениями между объектами и реальным программным кодом. Объекты помещаются в вашу форму, при этом код, отвечающий объектам, автоматически записывается в исходный файл. Этот код компилируется, обеспечивая существенно более высокую производительность, чем визуальная среда, которая интерпретирует информацию лишь в ходе исполнения программы.

### **3.2. Описание команд и ответов**

*Супервизорный формат* показан на рис. 2.1 и предусматривает четыре команды и ответа (RR, RNR, REJ, SREJ), которые представлены на рис. 2.4. (Обобщенная сводка всех команд и ответов приведена в таблице 3.1). Назначение этого формата состоит в выполнении нумерованных [т.е.

использующих порядковые номера кадров  $N(R)$ ] супервизорных функций, таких, как подтверждение (квитирование), опрос, временная задержка передачи данных и восстановление после ошибок. Кадры супервизорного формата не содержат информационного поля, следовательно, как показано на рис. 1, в них располагается только порядковый номер приема  $N(R)$ . Супервизорный формат может быть использован для подтверждения приема кадров от передающей станции.

Функции команд и ответов, используемых супервизорным форматом:

*RR (Receive ready – Готов к приему)* используется первичной или вторичной станцией для индикации того, что станция готова принять информационный кадр и / или подтвердить (квитировать) ранее принятые кадры с помощью поля  $N(R)$ . Если станция до этого, используя команду «Не готов к приему», посылала уведомление о том, что она занята, теперь она использует команду Готов к приему для индикации того, что она свободна и готова принять данные. Первичная станция может также использовать команду Готов к приему для опроса вторичной станции.

*RNR (Receive not ready – Не готов к приему)* используется станцией для индикации состояния занятости. Эта команда уведомляет передающую станцию о том, что принимающая станция не способна принять дополнительные поступающие данные. Кадр RNR, используя поле  $N(R)$ , может подтвердить прием ранее переданных кадров. Состояние занятости может быть сброшено посылкой кадра RR, а также некоторых других кадров, которые будут рассмотрены позднее.

*REJ (Reject – Неприем)* используется для запроса передачи кадров, начиная с кадра, указанного в поле  $N(R)$ . Подтверждаются все кадры с номерами до  $N(R) - 1$ . Кадр REJ может использоваться для реализации метода «Возвращение-на-N» (Go-Back-N).

*SREJ (Selective reject – Выборочный неприем)* используется станцией для запроса повторной передачи единственного кадра, который определен в поле  $N(R)$ . Как и в случае включающего подтверждения, подтверждение

распространяется на все информационные кадры с номерами до N(R) – 1 включительно. Выборочный неприем позволяет реализовать режим выборочного повторения. Как только передан кадр SREJ, следующие кадры принимаются и сохраняются для повторно передаваемого кадра.

*Ненумерованные команды и ответы* используются для отправки большинства индикаторов команд и ответов. Ненумерованные команды можно разбить на группы в соответствии с выполняемыми функциями:

- команды установки режима: SNRM, SARM, SABM, (SNRME, SARME, SABME - для расширенной адресации), SIM, RIM, DISC;
- команды передачи информации: UI, UP;
- команды восстановления: RSET;
- другие команды: XID, TEST, DM, UA, FRMR, RD.

Функции команд/ответов для ненумерованного формата:

*UI (Unnumbered information - Ненумерованная информация)*. Эта команда позволяет производить передачу данных пользователя в не нумерованном кадре (т. е. без порядкового номера).

*RIM (Request Initialization Mode - Режим инициализации запроса)*. Кадр RIM является запросом на команду SIM от вторичной к первичной станции.

*SIM (Set Initialization Mode - Установить режим инициализации)*. Эта команда используется для инициализации сеанса между первичной и вторичной станциями. Ожидаемым ответом является UA.

*SNRM (Set Normal Response Mode - Установить режим нормального ответа)*. Эта команда переводит вторичную станцию в NRM (режим нормального ответа). NRM предотвращает посылку вторичной станцией несанкционированных (unsolicited) кадров. Это означает, что первичная станция управляет всем потоком сообщений в канале.

*DM (Disconnect Mode - Режим разъединения)*. Этот кадр ответа передается вторичной станцией для индикации того, что она находится в режиме логического разъединения.

*DISC (Disconnect - Разъединить)*. Эта команда, передаваемая первичной станцией, переводит вторичную станцию в режим разъединения аналогично нажатию рычага телефонного аппарата.

*UA (Unnumbered Acknowledgment - Ненумерованное подтверждение)*. Это - положительное подтверждение АСК для установки режима команд (SIM, DISC, RESET). UA также используется для уведомления об окончании состояния занятости станции.

*FRMR (Frame Rejekt - Неприем кадра)*. Вторичная станция посылает этот кадр, когда она встречается с недействительным кадром. Причина указывается в информационном поле этого кадра. Кадр ответа FRMR используется при выполнении следующих условий:

- Прием недействительного управляющего поля команды или ответа.
- Прием слишком длинного информационного поля.
- Прием недействительного поля  $N(R)$ .
- Прием недопустимого информационного поля или супервизорного/ненумерованного кадра неправильной длины.

В информационном поле кадра FRMR содержится информация о состоянии (status)отвергнутого кадра:

- управляющее поле отвергнутого кадра;
- текущее значение переменных состояния принимающей станции - посылки  $V(S)$  и приема  $V(R)$ ;
- отвергнутый кадр был командой или ответом;
- управляющее поле является недействительным;
- кадр был передан с недопустимым информационным полем;
- информационное поле является слишком длинным;
- порядковые номера являются недействительными.

*RD (Request Disconnect - Запрос разъединения)*. Это запрос от вторичной станции на логическое разъединение и установление состояния логического разъединения.

*XID (Exchange State Identification - Идентификация станции при коммутации).*

Эта команда запрашивает идентификацию вторичной станции. В системах с коммутацией эта команда используется для идентификации вызывающей станции.

*TEST (Test-проверка).* Этот кадр (команда/ответ) используется для санкционирования тестовых ответов от вторичной станции. В ответе желательно сформировать информационную область, содержащуюся в команде.

*SARM (Set Asynchronous Response Mode - Установить режим асинхронных ответов).* Устанавливает режим, позволяющий вторичной станции вести передачу без опроса со стороны первичной станции. Он переводит вторичную станцию в состояние передачи информации (IS) режима ARM. Поскольку команда SARM устанавливает две несбалансированные станции, SARM должна выдаваться по обоим направлениям передачи:

- А посылает: B, DISC
- В посылает: B, UA A, DISC
- А посылает: A, UA
- В посылает: A, SARM
- А посылает: A, UA B, SARM
- В посылает: B, UA

Команды DISC посылаются, чтобы гарантировать полную реинициализацию канала.

*SABM (Set Asynchronous Balanced Mode - Установить асинхронный сбалансированный режим).* Устанавливает режим в ARM, в котором станции являются равноправными. Для передачи не требуется опроса, поскольку каждая станция является станцией комбинированного типа.

*SNRME (Set Normal Response Extended - Установить расширенный режим нормального ответа).* Устанавливает SNRM с двумя дополнительными байтами в управляющем поле.

*SABME (Set Asynchronous Balance Mode Extended - Установить расширенный асинхронный сбалансированный режим).* Устанавливает SABM с двумя дополнительными байтами в управляющем поле.

*UP (Unnumbered Poll - Ненумерованный опрос).* Команда опрашивает станцию безотносительно к нумерации кадров и квитированию. Если бит опроса установлен в 0, ответ является необязательной возможностью, предоставляемой только для одного ответа. В качестве ответа могут быть переданы неподтвержденные или еще непереданные I-кадры.

*RSET (Reset - Сброс переменных).* Передающая станция сбрасывает свой V(S), а принимающая станция свой V(R) с целью возобновления упорядоченной передачи в заданном направлении новой последовательности кадров. Эта команда используется для восстановления. Кадры, которые ранее не были квитированы, остаются таковыми.

Таблица 3.1 Обобщенная сводка всех команд и ответов

		Код управляющего поля									
	1	2	3	4	5	6	7	8	Команды	Ответы	
<b>I-формат</b>	0	N(S)			*	N(R)			I - Информация	I - Информация	
<b>S-формат</b>	1	0	0	0	*	N(R)			RR - Готов к приему	RR - Готов к приему	
	1	0	0	1	*	N(R)			REJ - Неприем	REJ - Неприем	
	1	0	1	0	*	N(R)			RNR - Не готов к приему	RNR - Не готов к приему	
	1	0	1	1	*	N(R)			SREJ - Выбор.	SREJ - Выбор.	

									неприем	Неприем
<b>U- формат</b>	1	1	0	0	*	0	0	0	UI - Ненумеров. информ.	UI - Ненумеров. информ.
	1	1	0	0	*	0	0	1	SNRM - Установить NRM	
	1	1	0	0	*	0	1	0	DISC - Разъединить	RD- Запрос разъед.
	1	1	0	0	*	1	0	0	UP- Ненумеров. опрос	
	1	1	0	0	*	1	1	0		UA - Ненумеров.подтв.
	1	1	0	0	*	1	1	1	TEST - Проверка	TEST - Проверка
	1	1	1	0	*	0	0	0	SIM - Уст.реж.иниц.	RIM - Запр.реж.иниц.
	1	1	1	0	*	0	0	1		FRMR - Неприем кадра
	1	1	1	1	*	0	0	0	SARM - Установить ARM	DM - Режим разъед.
	1	1	1	1	*	0	0	1	RSET - Сбросить	
1	1	1	1	*	0	1	0	SARME -		

									Уст.расш.ARM	
	1	1	1	1	*	0	1	1	SNRME - Уст.расш.NRM	
	1	1	1	1	*	1	0	0	SABM -установить ABM<TD&NBSP;< TD>	
	1	1	1	1	*	1	0	1	XID - Идентиф. станции	XID - Идентиф. станции
	1	1	1	1	*	1	1	0	SABME -Уст.расш. ABM	

### 3.3 Описание алгоритма

HDLC является *кодопрозрачным протоколом*. Он не зависит от конкретного кода (ASCII/IA5 или EBCDIC) при выполнении функции управления каналом. Восьмибитовая комбинация флага 01111110 помещается в начале и в конце кадра, чтобы дать возможность приемнику распознать начало и конец кадра. Возможны случаи, когда прикладной процесс помещает в данных пользователя последовательность 01111110, совпадающую с флагом. В этом случае передающая станция в поток выходных данных помещает 0 после 5 подряд идущих единиц, встретившихся в любом месте между начальным и конечным флагами кадра. Такая вставка производится в адресное, управляющее, информационное поля и поле CRC. Этот метод называется вставкой битов (*bit stuffing*); такую же функцию выполняет знак DLE в протоколе BSC. После того как завершается

вставка битов в кадр и по концам кадра помещаются флаги, кадр передается приемнику по каналу.

Приемник постоянно контролирует поток битов. При получении нуля с пятью далее идущими подряд единицами (011111) анализирует следующий (седьмой) бит. Если это нуль, он удаляет этот бит. Однако если седьмой бит является единицей (0111111), приемник анализирует восьмой бит. Если это нуль (01111110), он считает, что получена флаговая комбинация. Если это единица, выполняется анализ последующих бит. Возможна ситуация приема либо сигнала покоя, либо сигнала аварийного завершения, на которые станция реагирует соответствующим образом. Таким образом, в протоколе HDLC обеспечиваются кодовая прозрачность по данным. Протоколу безразлично, какие кодовые комбинации находятся в потоке данных. Единственное, что требуется, – это поддерживать уникальность флагов.

HDLC используется также два других сигнала: *сигнал аварийного завершения (АЗ)* состоит из последовательности единиц, число которых не меньше семи и не больше четырнадцати; *состояние покоя* представляется последовательностью пятнадцати или большего числа единиц.

*Управление потоком* в HDLC осуществляется с помощью передающих и принимающих окон. Окно устанавливается на каждом конце канала связи, чтобы обеспечить резервирование ресурсов обеих станций. Этими ресурсами могут быть ресурсы вычислителя или пространство буфера. В большинстве случаев окно обеспечивает и буферное пространство, и правила нумерации (сообщений). Окно устанавливается во время инициирования сеанса связи между станциями. Если станция А и станция В должны обмениваться данными, А резервирует окно для В, а В резервирует окно для А. Использование окон необходимо для полnodуплексных протоколов, потому что они подразумевают непрерывный поток кадров в принимающий узел без периодических подтверждений с остановкой и ожиданием.

*Переменные состояния станции  $V(S)$  и  $V(R)$* . Окна в принимающем и передающем узлах управляются переменными состояниями, которые

представляют по сути состояние счетчика. Передающий узел поддерживает переменную состояния посылки  $V(S)$ . Это порядковый номер следующего по очереди I-кадра, который должен быть передан. Принимающий узел поддерживает переменную состояния приема  $V(R)$ , которая содержит номер, который, как ожидается, является порядковым номером следующего I-кадра.  $V(S)$  увеличивается на 1 при передаче каждого кадра и помещается в поле порядкового номера посылки кадра. Получив кадр, принимающий узел производит проверку наличия ошибок передачи и сравнивает порядковый номер со своим  $V(R)$ . Если кадр может быть принят, узел увеличивает  $V(R)$  на 1, помещает его в поле порядкового номера приема кадра подтверждения АСК и посылает этот кадр в узел-отправитель, завершая квитирование передачи.

Если  $V(R)$  не равен порядковому номеру посылки в кадре или обнаружена ошибка, значит, что-то произошло, и после тайм-аута в узел-отправитель посылается NAK [с порядковым номером приема, содержащим значение  $V(R)$ ]. В большинстве протоколов этот NAK называется Неприем (REJ) или Выборочный неприем (SREJ). Значение  $V(R)$  уведомляет передающее устройство ООД о том, что ожидается посылка нового кадра. Т. к. передатчик восстанавливает старое значение  $V(S)$  и повторяет передачу кадра, порядковый номер которого совпадает со значением  $V(S)$ .

Во многих системах для  $V(S)$  и  $V(R)$  у порядковых номеров в кадре используются числа 0–7. если переменные состояния в результате последовательного увеличения достигли 7, то, начиная с 0, эти числа снова используются. Вследствие повторного использования чисел устройствам станциям не разрешено посылать кадр с порядковым номером, который не был подтвержден. Например, протокол должен дождаться подтверждения кадра с номером 6, прежде чем он опять использует  $V(S)=6$ . Этот процесс показан на рис. 1. Здесь кадры с 6 по 4 еще не подтверждены. Если бы был послан еще один кадр с порядковым номером 6, соответствующее

подтверждение АСК с номером 6 не позволило бы определить, приход какого кадра с порядковым номером 6 подтверждается.

Использование номеров 0–7 позволяет семи кадрам быть в активном состоянии, прежде, чем «закроется» окно. Несмотря на то что диапазон 0–7 дает восемь порядковых номеров,  $V(R)$  содержит значение следующего ожидаемого кадра, что ограничивает число активных кадров до 7.

### **3.4 Системные параметры T1, N2, N1, K и рекомендации по их установке**

*Таймер T1* запускается с момента передачи каждого кадра и используется для инициирования повторной передачи, в случае его переполнения. При выборе периода таймера T1 необходимо учитывать, запускается ли таймер по началу или по концу кадра. Для правильной работы процедуры необходимо, чтобы период таймера T1 был больше, чем максимальное время между передачей некоторого кадра (SARM, SABM, DM, DISC, FRMR, I или супервизорной команды) и приемом соответствующего кадра, возвращаемого в качестве отклика на этот кадр (UA, DM или подтверждающий кадр).

*Счетчик N2* используется для определения максимального числа повторных передач, выполняемых по переполнении таймера T1. Переменные T1 и N2 используются также командами / ответами установления звена, такими, как SABM и UA.

*Счетчик N1* - максимальное число битов в I-кадре. Определяет максимальную длину информационных полей.

*Размер окна K* - максимальное число переданных, но не подтвержденных I-кадров. Это максимальное число последовательно занумерованных I-кадров, которые в любой момент времени станции могут передать без получения подтверждения. Оно не должно быть более 7. Параметры T1, N2, N1 и K являются системными, подлежащими согласованию с администрацией на некоторый период времени.

### 3.5 Листинг программы

```
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ScktComp, ComCtrls, Buttons, ExtCtrls;
type
TForm1 = class(TForm)
GroupBox1: TGroupBox;
Edit1: TEdit;
Label1: TLabel;
Edit2: TEdit;
Label2: TLabel;
Edit3: TEdit;
Label3: TLabel;
Memo1: TMemo;
Label4: TLabel;
Edit5: TEdit;
Label5: TLabel;
Edit6: TEdit;
Label7: TLabel;
GroupBox2: TGroupBox;
RichEdit1: TRichEdit;
OpenDialog1: TOpenDialog;
Button1: TSpeedButton;
Button3: TSpeedButton;
SpeedButton1: TSpeedButton;
```

```
LabeledEdit1: TLabelledEdit;
RadioButton1: TRadioButton;
RadioButton2: TRadioButton;
RadioGroup1: TRadioGroup;
RadioGroup2: TRadioGroup;
Memo2: TMemo;
SpeedButton2: TSpeedButton;
GroupBox3: TGroupBox;
Memo3: TMemo;
procedure Button1Click (Sender: TObject);
procedure Button3Click (Sender: TObject);
procedure SpeedButton1Click (Sender: TObject);
procedure FormCreate (Sender: TObject);
procedure SpeedButton2Click (Sender: TObject);
private
  {Private declarations}
public
  {Public declarations}
end;
var
  Form1: TForm1;
  kl:boolean;
implementation
uses Unit2;
  {$R *.dfm}
procedure perez1016 (sr:string; var se:string);
var chis:real;
begin
  chis:=strtofloat(sr);
  se:=inttohex (trunc(chis), 24);
```

```

end;
procedure perev210 (sr:string; var se:string);
var j, chis, i, pol:integer;
begin
chis:=0;
pol:=length(sr);
j:=0;
for i:=pol downto 1 do
begin
if sr[i]='1' then
chis:=chis+trunc (exp(j*ln(2)));
j:=j+1;
end;
se:=inttostr(chis);
end;
procedure perev162 (sr:string; var se:string);
var chis:real; i, p:integer; so:string;
begin
p:=length(sr);
se:="";
for i:=1 to p do
begin
case sr[i] of
'0':se:=se+'0000';
'1':se:=se+'0001';
'2':se:=se+'0010';
'3':se:=se+'0011';
'4':se:=se+'0100';
'5':se:=se+'0101';
'6':se:=se+'0110';

```

```

'7':se:=se+'0111';
'8':se:=se+'1000';
'9':se:=se+'1001';
'A':se:=se+'1010';
'B':se:=se+'1011';
'C':se:=se+'1100';
'D':se:=se+'1101';
'E':se:=se+'1110';
'F':se:=se+'1111';
end;
end;
end;
procedure TForm1. Button1Click (Sender: TObject);
var st:string;
begin
if InputQuery ('Введите IP адрес первичной станции', 'Адрес вторичной
станции', st) then
begin
edit2. Text:='1010010101';
end;
end;
procedure Delay (msecs: Longint);
var
FirstTick: Longint;
begin
FirstTick:= GetTickCount;
repeat
Application. ProcessMessages;
until GetTickCount – FirstTick >= msecs;
end;

```

```

procedure TForm1. SpeedButton1Click (Sender: TObject);
begin
if opendialog1. Execute then
begin
kl:=true;
LabeledEdit1. Text:=OpenDialog1. FileName;
memo3. Lines. LoadFromFile (OpenDialog1. FileName);
end;
end;

procedure TForm1. Button3Click (Sender: TObject);
var st6, st5, st3, st2, st1, st, st4:string; kol, k, i, j, im:integer; f:file of char;
ch:char;
begin
if kl=true then
begin
kl:=false;
assignfile (f, OpenDialog1.filename);
reset(f);
form2. Memo3. Lines. Clear;
while not eof(f) do
begin
im:=0;
form2. Memo2. Lines. Clear;
edit3. Text:='10100000';
memo2. Lines. Add ('Команда RR – готов к приему');
memo2. Lines. Add ('-');
Radiogroup1.itemindex:=0;
RichEdit1. Lines. Clear;
RichEdit1.
Lines.
Add('011111101010010101100000000000000001111110');

```

```

RadioGroup2. ItemIndex:=1;
delay(2000);
form2. RadioGroup1. ItemIndex:=0;
form2. RadioGroup2. ItemIndex:=1;
form2.memo1. Lines. Clear;
form2.memo1. Lines.
Add('0111110101001010110000000000000000001111110');
form2. Show;
delay(2000);
form2. Memo1. Lines. Clear;
form2. Memo1. Lines.
Add('0111110101001010110000000000000000001111110');
form2. RadioGroup1.itemindex:=1;
form2. RadioGroup2. ItemIndex:=1;
delay(2000);
form2. Hide;
edit3. Text:='10101000';
memo2. Lines. Add ('Ответ RR – ГОТОВ К ПРИЕМУ');
memo2. Lines. Add ('-');
RichEdit1. Lines. Clear;
RichEdit1. Lines.
Add('0111110101001010110000000000000000001111110');
RadioGroup1.itemindex:=1;
RadioGroup2. ItemIndex:=1;
st:="";
st:='01111101010010101';
i:=0;
st3:="";
RichEdit1. Lines. Clear;
memo2. Lines. Add('Передача');

```

```

memo2. Lines. Add ('-');
st4:="";
st4:=st4;
st5:="";
st6:="";
while (not eof(f)) and (i<=512) do
begin
read (f, ch);
st6:=st6+ch;
perv1016 (inttostr(ord(ch)), st3);
perv162 (st3, st3);
st4:=st4+st3;
kol:=0;
for k:=1 to length(st3) do
begin
if st3 [k]='1' then kol:=kol+1;
if kol=6 then begin insert ('0', st3, k); kol:=0; end
end;
i:=i+1;
st5:=st5+st3;
end;
memo1. Lines. Clear;
memo1. Lines. Add(st4);
//closefile(f);
st5:=st5+'0000000000000000001111110';
RichEdit1. Lines. Add(st5);
RadioGroup2. ItemIndex:=0;
edit3. Text:='00000000';
delay(2000);
/////

```

```
form2. RadioGroup1. ItemIndex:=0;
form2. RadioGroup2. ItemIndex:=0;
form2.memo1. Lines. Clear;
form2.memo1. Lines. Add(st5);
form2. Show;
delay(2000);
form2. Memo2. Lines. Clear;
form2. Memo2. Lines. Add(st4);
form2. RadioGroup1.itemindex:=0;
form2. RadioGroup2. ItemIndex:=0;
delay(2000);
form2. Memo3. Lines. Add(st6);
form2. RadioGroup1.itemindex:=0;
form2. RadioGroup2. ItemIndex:=0;
delay(2000);
form2. Hide;
end;
closefile(f);
//ClientSocket1. Socket. SendText ('#End');
MessageDlg ('Передача данных закончилась', mtInformation, [mbOk], 0);
end
else
MessageDlg ('Введите информацию', mtError, [mbOk], 0);
end;
procedure TForm1. FormCreate (Sender: TObject);
begin
kl:=false;
end;
procedure TForm1. SpeedButton2Click (Sender: TObject);
begin
```

halt;  
end;  
end.

## **ВЫВОДЫ**

В данной главе рассмотрено имитационное моделирование протокола HDLC. Потребность в разработке и применении эффективных и адекватных реальной действительности компьютерных программ и технологий сегодня возрастает. Компьютерная технология незаменима, поскольку она даёт возможность оптимизировать и рационализировать управленческую функцию за счет применения новых средств сбора, передачи и преобразования информации. Delphi обладает широким набором возможностей, начиная от проектировщика форм и кончая поддержкой всех форматов популярных баз данных. Имитационное моделирование разработана - написана в программной среде Delphi.

## **4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ**

### **4.1 Гиподинамия и их влияние на организм человека**

Снижение физических нагрузок в условиях современной жизни, с одной стороны, и недостаточное развитие массовых форм физической культуры среди населения, с другой стороны, приводят к ухудшению различных функций и появлению негативных состояний организма человека. Понятия гипокинезия и гиподинамия. Для обеспечения нормальной жизнедеятельности организма человека необходима достаточная активность скелетных мышц. Работа мышечного аппарата способствует развитию мозга и установлению меж центральных и межсенсорных взаимосвязей.

Двигательная деятельность повышает энергопродукцию и образование тепла, улучшает функционирование дыхательной, сердечнососудистой и других систем организма. Недостаточность движений нарушает нормальную работу всех систем и вызывает появление особых состояний – гипокинезии и гиподинамии. Гипокинезия – это пониженная двигательная активность. Она может быть связана с физиологической незрелостью организма, с особыми условиями работы в ограниченном пространстве, с некоторыми заболеваниями и др. причинами. В некоторых случаях (гипсовая повязка, постельный режим) может быть полное отсутствие движений или акинезия, которая переносится организмом еще тяжелее. Существует и близкое понятие — гиподинамия. Это понижение мышечных усилий, когда движения осуществляются, но при крайне малых нагрузках на мышечный аппарат. В обоих случаях скелетные мышцы нагружены совершенно недостаточно. Возникает огромный дефицит биологической потребности в движениях, что резко снижает функциональное состояние и работоспособность организма. Некоторые животные очень тяжело переносят отсутствие движений. Например, при содержании крыс в течение 1 месяца в условиях акинезии выживает 60% животных, а в условиях гипокинезии – 80%. Цыплята, выращенные в условиях обездвижения в тесных клетках и выпущенные затем на волю, погибали при малейшей пробежке по двору. Тяжело переносится снижение двигательной активности человеком. Обследование моряков подводников показало, что после 1,5 месяцев пребывания в море сила мышц туловища и конечностей уменьшалась на 20-40% от исходной, а после 4 месяцев плавания – на 40-50%.

Еще в древности было замечено, что физическая активность способствует формированию сильного и выносливого человека, а неподвижность ведет к снижению работоспособности, заболеваниям и тучности. Все это происходит вследствие нарушения обмена веществ. Уменьшение энергетического обмена, связанное с изменением

интенсивности распада и окисления органических веществ, приводит к нарушению биосинтеза, а также к изменению кальциевого обмена в организме. Вследствие этого в костях происходят глубокие изменения. Прежде всего, они начинают терять кальций. Это приводит к тому, что кость делается рыхлой, менее прочной. Кальций попадает в кровь, оседает на стенках кровеносных сосудов, они склерозируются, т. е. пропитываются кальцием, теряют эластичность и становятся ломкими. Способность крови к свертыванию резко возрастает. Возникает угроза образования кровяных сгустков (тромбов) в сосудах. Содержание большого количества кальция в крови способствует образованию камней в почках. Отсутствие мышечной нагрузки снижает интенсивность энергетического обмена, что отрицательно сказывается на скелетных и сердечной мышцах. Кроме того, малое количество нервных импульсов, идущих от работающих мышц, снижает тонус нервной системы, утрачиваются приобретенные ранее навыки, не образуются новые.

Все это самым отрицательным образом отражается на здоровье. Следует учесть также следующее. Сидячий образ жизни приводит к тому, что хрящ постепенно становится менее эластичным, теряет гибкость. Это может повлечь снижение амплитуды дыхательных движений и потерю гибкости тела. Но особенно сильно от неподвижности или малой подвижности страдают суставы. Характер движения в суставе определен его строением. В коленном суставе ногу можно только сгибать и разгибать, а в тазобедренном суставе движения могут совершаться во всех направлениях. Однако амплитуда движений зависит от тренировки. При недостаточной подвижности связки теряют эластичность. В полость сустава при движении выделяется недостаточное количество суставной жидкости, играющей роль смазки. Все это затрудняет работу сустава. Недостаточная нагрузка влияет и на кровообращение в суставе. В результате питание костной ткани нарушается, формирование суставного хряща, покрывающего головку и суставную впадину сочленяющихся костей, да и самой кости идет

неправильно, что приводит к различным заболеваниям. Но дело не ограничивается только этим. Нарушение кровообращения может привести к неравномерному росту костной ткани, вследствие чего возникает разрыхление одних участков и уплотнение других. Форма костей в результате этого может стать неправильной, а сустав потерять подвижность.

## **4.2. Пожарная безопасность**

Пожары наносят громадный материальный ущерб и в ряде случаев сопровождаются гибелью людей. Поэтому защита от пожаров является важнейшей обязанностью каждого члена общества и проводится в общегосударственном масштабе. Противопожарная защита имеет своей целью изыскание наиболее эффективных, экономически целесообразных и технически обоснованных способов, средств предупреждения пожаров, и их ликвидации с минимальным ущербом при наиболее рациональном использовании сил и технических средств тушения.

Пожарная безопасность – это состояние объекта, при котором исключается возможность пожара, а в случае его возникновения используются необходимые меры по устранению негативного влияния опасных факторов пожара на людей, сооружения и материальных ценностей. Пожарная безопасность может быть обеспечена мерами пожарной профилактики и активной пожарной защиты. Пожарная профилактика включает комплекс мероприятий, направленных на предупреждение пожара или уменьшение его последствий.

Почти во всех производствах применяются вещества, способные воспламеняться и гореть, а в некоторых случаях - образовывать с воздухом взрывоопасные смеси. Горение – быстротекущая реакция окисления, сопровождающаяся выделением тепла и (обычно) света. Химическая реакция горения всегда является сложной и состоит из ряда элементарных

химических превращений. Химическое превращение при горении протекает одновременно с физическими процессами: переносом тепла и массы. Поэтому скорость горения всегда определяется как условиями тепло и массопередачи, так и скоростью протекания химических превращений.

Для возникновения горения необходимо наличие: горючего вещества, окислителя и импульса. Импульсом может быть: открытый огонь, искра (электрическая, статическая или от удара металлических предметов, молния, нагрев вещества выше температуры его самовоспламенения и др.). Горючие вещества бывают в трех агрегатных состояниях: твердом, жидком и газообразном (возможно и 4-ое состояние вещества - плазма). При горении твердых материалов горючее вещество и воздух не перемешаны, имеют поверхность раздела, и горение протекает в так называемом диффузионном режиме, т.е. скорость реакции определяется скоростью подвода (отвода) продуктов реакции (лимитирующая стадия - диффузия).

Если молекулы кислорода хорошо перемешаны с горючим веществом - горение определяется кинетикой химической реакции (обмен электронами), а режим - кинетическим. Горение такой смеси может происходить в виде взрыва.

Причинами взрывов и пожаров могут быть не только халатное и небрежное обращение с открытым огнем, но и ошибки в проектировании, нарушение технологического процесса, неисправность, перегрузка или неправильное устройство электрических сетей, производственного оборудования, разряды статического электричества, неисправность установок и систем.

Пожар – это горение вне специального очага, которое не контролируется и может привести к массовому поражению и гибели людей, а также к нанесению экологического, материального и другого вреда.

Горение - это химическая реакция окисления, сопровождающаяся выделением теплоты и света. Для возникновения горения требуется наличие трех факторов: горючего вещества, окислителя и источника загорания.

Окислителями могут быть кислород, хлор, фтор, бром, йод, окиси азота и другие. Кроме того, необходимо чтобы горючее вещество было нагрето до определенной температуры и находилось в определенном количественном соотношении с окислителем, а источник загорания имел определенную энергию.

Наибольшая скорость горения наблюдается в чистом кислороде. При уменьшении содержания кислорода в воздухе горение прекращается. Горение при достаточной концентрации окислителя называется полным, а при его нехватке – неполным.

Выделяют три основных вида самоускорения химической реакции при горении: тепловой, цепной и цепочно-тепловой. Тепловой механизм связан с экзотермичностью процесса окисления и возрастанием скорости химической реакции с повышением температуры. Цепное ускорение реакции связано с катализом превращений, которое осуществляют промежуточные продукты превращений. Реальные процессы горения осуществляются, как правило, по комбинированному (цепочно-тепловой) механизму.

Процесс возникновения горения подразделяется на несколько видов. Вспышка - быстрое сгорание горючей смеси, не сопровождающееся образованием сжатых газов. Возгорание - возникновение горения под воздействием источника зажигания. Воспламенение - возгорание, сопровождающееся появлением пламени. Самовозгорание - явление резкого увеличения скорости экзотермических реакций, приводящее к возникновению горения вещества при отсутствии источника зажигания.

Самовоспламенение - самовозгорание, сопровождается появлением пламени. Взрыв - процесс чрезвычайно быстрого, под влиянием внешнего источника воспламенения, химического превращения вещества, сопровождающегося выделением газов и большого количества тепла, нагревающего эти газы до высокой температуры, в результате чего газы совершают работу. Взрывная способность горючих газов, паров и пыли в воздухе сохраняется в определенных интервалах их концентраций.

Существуют нижние и верхние концентрационные и температурные пределы распространения пламени. Нижний (верхний) концентрационные пределы распространения пламени - минимально (максимальное) содержание горючего вещества в однородной смеси с окислительной средой, при которой возможно распространение пламени по смеси на любое расстояние от источника зажигания. Невозможность воспламенения горючей смеси при концентрации ниже НКПРП объясняется малым количеством горючего вещества и избытком воздуха. Чем меньше коэффициент избытка воздуха, тем больше скорость горения и выше давление паров при взрыве.

Режим короткого замыкания — появление в результате резкого возрастания силы тока, электрических искр, частиц расплавленного металла, электрической дуги, открытого огня, воспламенившейся изоляции.

Причины возникновения короткого замыкания: ошибки при проектировании; старение изоляции; увлажнение изоляции; механические перегрузки.

Пожарная опасность при перегрузках — чрезмерное нагревание отдельных элементов, которое может происходить при ошибках проектирования в случае длительного прохождения тока, превышающего номинальное значение. При 1,5 кратном превышении мощности резисторы нагреваются до 200-300 С.

Пожарная опасность переходных сопротивлений — возможность воспламенения изоляции или др. горючих близлежащих материалов от тепла, возникающего в месте аварийного сопротивления (в переходных клеммах, переключателях и др.).

Пожарная опасность перенапряжения — нагревание токоведущих частей за счет увеличения токов, проходящих через них, за счет увеличения перенапряжения между отдельными элементами электроустановок. Возникает при выходе из строя или изменении параметров отдельных элементов.

Пожарная опасность токов утечки — локальный нагрев изоляции между отдельными токоведущими элементами и заземленными конструкциями.

Меры пожарной профилактики строительно-планировочные; технические; организационные.

Строительно-планировочные определяются огнестойкостью зданий и сооружений (выбор материалов конструкций: сгораемые, негораемые, трудно сгораемые) и предел огнестойкости — это количество времени, в течение которого под воздействием огня не нарушается несущая способность строительных конструкций вплоть до появления первой трещины.

Технические меры:

- это соблюдение противопожарных норм при эвакуации систем вентиляции, отопления, освещения, электрического обеспечения и т.д.
- использование разнообразных защитных систем;
- соблюдение параметров технологических процессов и режимов работы оборудования.

Организационные меры — проведение обучения по пожарной безопасности, соблюдению мер по пожарной безопасности.

Территория базы, склада, предприятия торговли и общественного питания должна постоянно содержаться в чистоте, а после окончания работы тщательно очищаться от упаковочного материала, отходов и горючего мусора. Отходы, упаковочные материалы необходимо систематически удалять на специально отведенные огражденные участки и своевременно вывозить.

Ко всем зданиям и сооружениям должен быть обеспечен свободный доступ. Проезды и подъезды к пожарным водоисточникам, а также подступы к пожарному инвентарю и оборудованию должны быть всегда свободными. Противопожарные разрывы между зданиями не разрешается использовать под складирование материалов, оборудования, упаковочной тары, стоянку транспортных средств.

В зимний период дороги, проезды, подъезды и крышки люков пожарных гидрантов и водоемов систематически очищают от льда и снега. Хранение товарно-материальных ценностей, тары на рампах складов не

допускается; материалы, разгруженные на рампу, к концу работы склада должны быть убраны.

О закрытии отдельных участков дорог или проездов для их ремонта (или по другим причинам), препятствующих проезду пожарных машин, руководитель предприятия или лицо, ответственное за противопожарное состояние объекта, обязан немедленно уведомить пожарную охрану.

На период производства работ по ремонту дорог на объекте в соответствующих местах устанавливают указатели направления объезда или устраивают переезды через ремонтируемые участки. Разводить костры, сжигать отходы, тару и упаковочные материалы на территории предприятия запрещается. Территория предприятия в ночное время должна освещаться.

На территории баз (складов) в сельской местности необходимо иметь приспособление для подачи сигналов о пожаре. Торговые, складские, производственные, административные, бытовые и другие помещения нужно постоянно содержать в чистоте и обеспечивать первичными средствами пожаротушения согласно нормам.

Устройства противопожарной защиты технологических и дверных проемов во внутренних стенах и междуэтажных перекрытиях (противопожарные двери, заслонки, шиберы, водяные завесы и т.п.) должны постоянно находиться в работоспособном состоянии. При пересечении противопожарных преград различными коммуникациями зазоры между ними и строительными конструкциями (на всю их толщину) не должны иметь неплотности, через которые могут проникать продукты горения.

Курение в складских и торговых помещениях и на их территории запрещается. Курить разрешается только в специально отведенных местах, обеспеченных средствами пожаротушения, урнами (ящиками с песком). Эти места должны иметь указательные знаки по ГОСТ 12.4.026-76. Наружные пожарные лестницы, а также ограждения безопасности на крышах зданий необходимо содержать в исправном состоянии.

Для использования обтирочных материалов устанавливаются металлические ящики с плотно закрывающимися крышками. После окончания работы ящики следует очищать от обтирочных материалов.

Спецодежда лиц, работающих с маслами, лаками, красками должна храниться в металлических шкафах, установленных в специально отведенных для этой цели местах.

## **ЗАКЛЮЧЕНИЕ**

Долгое время основным протоколом выделенных линий был протокол HDLC (High-level Data Link Control), имеющий статус стандарта ISO. Протокол HDLC на самом деле представляет собой семейство протоколов, в которое входят известные протоколы: LAP-B, образующий канальный уровень сетей X.25, LAP-D - канальный уровень сетей ISDN, LAP-M - канальный уровень асинхронно-синхронных модемов, LAP-F - канальный уровень сетей frame relay.

Протокол канального HDLC обеспечивает следующие функции:

- управление передачей данных через физический канал организованный на первом уровне;
- проверка информационного канала;
- формирование кадра т.е. окаймление передаваемых данных служебными символами данного уровня;

- контроль данных;
- обеспечение прозрачности информационного канала;
- управление каналом передачи данных;

В данной моей работе с помощью программы Delphi рассмотрено визуальное представление работы протокола HDLC. Размещение объектов в Delphi связано с более тесными отношениями между объектами и реальным программным кодом. Визуальное программирование как бы добавляет новое измерение при создании приложений, давая возможность изображать эти объекты на экране монитора до выполнения самой программы. Без визуального программирования процесс отображения требует написания фрагмента кода, создающего и настоящего объект «по месту». Увидеть закодированные объекты было возможно только в ходе исполнения программы. При таком подходе достижение того, чтобы объекты выглядели и вели себя заданным образом, становится утомительным процессом, который требует неоднократных исправлений программного кода с последующей прогонкой программы и наблюдения за тем, что в итоге получилось.

Благодаря средствам визуальной разработки можно работать с объектами, держа их перед глазами и получая результаты практически сразу. Способность видеть объекты такими, какими они появляются в ходе исполнения программы, снимает необходимость проведения множества операций вручную, что характерно для работы в среде, не обладающей визуальными средствами – вне зависимости от того, является она объектно-ориентированной или нет. После того, как объект помещен в форму среды визуального программирования, все его атрибуты сразу отображаются в виде кода, который соответствует объекту как единице, исполняемой в ходе работы программы, как видно из выполненной работы.

Семейство HDLC протоколов канального уровня продолжает развиваться и расширяться. Однако акцент на обеспечение безошибочной передачи ослабляется по мере использования высокоскоростных каналов

связи и использования методов коррекции ошибок. В полной мере возможности HDLC протокола используется при построении глобальных сетей передачи данных.

### **ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА**

1. Блэк Ю. Сети ЭВМ: протоколы, стандарты, интерфейсы. М., Мир, 1990.
2. Протоколы информационно-вычислительных сетей: Справочник/ С.А. Аничкин, С.А. Белов, А.В. Берштейн и др.; Под. ред. И.А Мизина, А.П. Кулешова. – М.: Радио и связь, 1990. – 504 с. ил.
3. Бесстужев И.Н. Организация локальных сетей на базе персональных компьютеров. М.: СК Пресс, 1995.
4. COMPUTERCLUB ежемесячный журнал по компьютерам и телекоммуникациям. Ноябрь, №11, 1999
5. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов / В.Г. Олифер, Н.А. Олифер. – 3-е изд. – СПб: Питер, 2006 – 957 с.
6. Халсалл, Фред. Передача данных, сети компьютеров и взаимосвязь открытых систем Ф. Халсалл; Пер. с англ. Т. М. Тер-Микаэляна – М.: Радио и связь, 1995 – 408 с.

7. Гук, М. Аппаратные средства локальных сетей: энцикл. / М. Гук, Гл. ред. Е. Строганова. – СПб.: Питер, 2001. – 572 с.
8. <http://cp.ru/goods/common/html/tech/cookies.html>
9. [http://www.netscape.com/newsfer/std/cookie\\_spec.html](http://www.netscape.com/newsfer/std/cookie_spec.html)
10. <http://www.citforum.ru> – описание компьютерных технологий, аналитическая информация
11. <http://astu.secna.ru/russian/students/personal/>