Федеральное агенство связи

Сибирский государственный университет телекоммуникаций и информатики

Уральскийтехнический институт связи и информатики (филиал)

Индивидуальное задание №9

На тему: «Назначение ЭМ ВОС. Методы доступа в ЛВС.

Стандарт Сети Token Ring»

Выполнил: студент гр. 781

Бабушкин А.В.

Проверил: Тюпина О.М.

Екатеринбург 2010

СОДЕРЖАНИЕ

Назначение ЭМ ВОС и всех уровней протокольного стека

**Сетевая модель OSI** (базовая эталонная модель взаимодействия открытых систем, англ. *Open Systems Interconnection Basic Reference Model*) — абстрактная сетевая модель для коммуникаций и разработки сетевых протоколов. Представляет уровневый подход к компьютерной сети. Каждый уровень обслуживает свою часть процесса взаимодействия. Благодаря такой структуре совместная работа сетевого оборудования и программного обеспечения становится гораздо проще и прозрачнее.

Модель состоит из 7-ми уровней, расположенных друг над другом. Уровни взаимодействуют друг с другом (по «вертикали») посредством интерфейсов, и могут взаимодействовать с параллельным уровнем другой системы (по «горизонтали») с помощью протоколов. Каждый уровень может взаимодействовать только со своими соседями и выполнять отведённые только ему функции.

|  |
| --- |
| **Модель OSI** |
| **Тип данных** | **Уровень** | **Функции** |
| Данные | 7. Прикладной уровень | Доступ к сетевым службам |
| 6. Уровень представления | Представление и кодирование данных |
| 5. Сеансовый уровень | Управление сеансом связи |
| Сегменты | 4. Транспортный | Прямая связь между конечными пунктами и надежность |
| Пакеты | 3. Сетевой | Определение маршрута и логическая адресация |
| Кадры | 2. Канальный | Физическая адресация |
| Биты | 1. Физический  | Работа со средой передачи, сигналами и двоичными данными |

**Прикладной (Приложений) уровень (*англ.* *Application layer*)**

Верхний уровень модели, обеспечивает взаимодействие пользовательских приложений с сетью. Этот уровень позволяет приложениям использовать сетевые службы, такие как удалённый доступ к файлам и базам данных, пересылка электронной почты. Также отвечает за передачу служебной информации, предоставляет приложениям информацию об ошибках и формирует запросы к уровню представления. Пример: HTTP, POP3, SMTP, FTP, XMPP, OSCAR, Modbus, SIP, TELNET

**Представительский (Уровень представления) (*англ.* *Presentation layer*)**

Этот уровень отвечает за преобразование протоколов и кодирование/декодирование данных. Запросы приложений, полученные с прикладного уровня, он преобразует в формат для передачи по сети, а полученные из сети данные преобразует в формат, понятный приложениям. На этом уровне может осуществляться сжатие/распаковка или кодирование/декодирование данных, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально.

Уровень 6 (представлений) эталонной модели OSI обычно представляет собой промежуточный протокол для преобразования информации из соседних уровней. Это позволяет осуществлять обмен между приложениями на разнородных компьютерных системах прозрачным для приложений образом. Уровень представлений обеспечивает форматирование и преобразование кода. Форматирование кода используется для того, чтобы гарантировать приложению поступление информации для обработки, которая имела бы для него смысл. При необходимости этот уровень может выполнять перевод из одного формата данных в другой. Уровень представлений имеет дело не только с форматами и представлением данных, он также занимается структурами данных, которые используются программами. Таким образом, уровень 6 обеспечивает организацию данных при их пересылке.

Чтобы понять, как это работает, представим, что имеются две системы. Одна использует для представления данных расширенный двоичный код обмена информацией EBCDIC, например, это может быть мэйнфрейм компании IBM, а другая — американский стандартный код обмена информацией ASCII (его используют большинство других производителей компьютеров). Если этим двум системам необходимо обменяться информацией, то нужен уровень представлений, который выполнит преобразование и осуществит перевод между двумя различными форматами.

Другой функцией, выполняемой на уровне представлений, является шифрование данных, которое применяется в тех случаях, когда необходимо защитить передаваемую информацию от приема несанкционированными получателями. Чтобы решить эту задачу, процессы и коды, находящиеся на уровне представлений, должны выполнить преобразование данных. На этом уровне существуют и другие подпрограммы, которые сжимают тексты и преобразовывают графические изображения в битовые потоки, так что они могут передаваться по сети.

Стандарты уровня представлений также определяют способы представления графических изображений. Для этих целей может использоваться формат PICT — формат изображений, применяемый для передачи графики QuickDraw между программами для компьютеров Macintosh и PowerPC. Другим форматом представлений является тэгированный формат файлов изображений TIFF, который обычно используется для растровых изображений с высоким разрешением. Следующим стандартом уровня представлений, который может использоваться для графических изображений, является стандарт, разработанный Объединенной экспертной группой по фотографии (Joint Photographic Expert Group); в повседневном пользовании этот стандарт называют просто JPEG.

Существует другая группа стандартов уровня представлений, которая определяет представление звука и кинофрагментов. Сюда входят интерфейс электронных музыкальных инструментов MIDI (Musical Instrument Digital Interface) для цифрового представления музыки, разработанный Экспертной группой по кинематографии стандарт MPEG, используемый для сжатия и кодирования видеороликов на компакт-дисках, хранения в оцифрованном виде и передачи со скоростями до 1,5 Мбит/с, и QuickTime — стандарт, описывающий звуковые и видео элементы для программ, выполняемых на компьютерах Macintosh и PowerPC.

Пример: AFP - Apple Filing Protocol, ASCII - American Standard Code for Information Interchange, EBCDIC - Extended Binary Coded Decimal Interchange Code, ICA - Independent Computing Architecture, LPP - Lightweight Presentation Protocol, NCP - NetWare Core Protocol, NDR - Network Data Representation RDP - Remote Desktop Protocol, XDR - eXternal Data Representation, X.25 PAD - Packet Assembler/Disassembler Protocol

**Сеансовый уровень (*англ.* *Session layer*)**

5-й уровень модели отвечает за поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время. Уровень управляет созданием/завершением сеанса, обменом информацией, синхронизацией задач, определением права на передачу данных и поддержанием сеанса в периоды неактивности приложений. Синхронизация передачи обеспечивается помещением в поток данных контрольных точек, начиная с которых возобновляется процесс при нарушении взаимодействия.

Пример: ADSP (AppleTalk Data Stream Protocol), ASP (AppleTalk Session Protocol), H.245 (Call Control Protocol for Multimedia Communication), ISO-SP (OSI Session Layer Protocol (X.225, ISO 8327)), iSNS (Internet Storage Name Service), L2F (Layer 2 Forwarding Protocol), L2TP (Layer 2 Tunneling Protocol), NetBIOS (Network Basic Input Output System), PAP (Password Authentication Protocol), PPTP (Point-to-Point Tunneling Protocol), RPC (Remote Procedure Call Protocol), RTCP (Real-time Transport Control Protocol), SMPP (Short Message Peer-to-Peer), SCP (Secure Copy Protocol), ZIP (Zone Information Protocol), SDP (Sockets Direct Protocol).

**Транспортный уровень (англ. Transport layer)**

4-й уровень сетевой модели OSI предназначен для доставки данных без ошибок, потерь и дублирования в той последовательности, как они были переданы. При этом не важно, какие данные передаются, откуда и куда, то есть он предоставляет сам механизм передачи. Блоки данных он разделяет на фрагменты, размер которых зависит от протокола, короткие объединяет в один, а длинные разбивает. Протоколы этого уровня предназначены для взаимодействия типа точка-точка. Пример: TCP, UDP.

Существует множество классов протоколов транспортного уровня, начиная от протоколов, предоставляющих только основные транспортные функции (например, функции передачи данных без подтверждения приема), и заканчивая протоколами, которые гарантируют доставку в пункт назначения нескольких пакетов данных в надлежащей последовательности, мультиплексируют несколько потоков данных, обеспечивают механизм управления потоками данных и гарантируют достоверность принятых данных.

Некоторые протоколы сетевого уровня, называемые протоколами без установки соединения, не гарантируют, что данные доставляются по назначению в том порядке, в котором они были посланы устройством-источником. Некоторые транспортные уровни справляются с этим, собирая данные в нужной последовательности до передачи их на сеансовый уровень. Мультиплексирование (multiplexing) данных означает, что транспортный уровень способен одновременно обрабатывать несколько потоков данных (потоки могут поступать и от различных приложений) между двумя системами. Механизм управления потоком данных — это механизм, позволяющий регулировать количество данных, передаваемых от одной системы к другой. Протоколы транспортного уровня часто имеют функцию контроля доставки данных, заставляя принимающую данные систему отправлять подтверждения передающей стороне о приеме данных.

Пример: ATP (AppleTalk Transaction Protocol), CUDP (Cyclic UDP), DCCP (Datagram Congestion Control Protocol), FCP (Fiber Channel Protocol), IL (IL Protocol), NBF (NetBIOS Frames protocol), NCP (Netware Core Protocol), SCTP (Stream Control Transmission Protocol), SPX (Sequenced Packet Exchange), SST (Structured Stream Transport), TCP (Transmission Control Protocol), UDP (User Datagram Protocol).

**Сетевой уровень (англ. *Network layer*)**

Протокол сетевого уровня (англ. Network layer) — протокол 3-его уровня сетевой модели OSI, предназначается для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию, отслеживание неполадок и заторов в сети. На этом уровне работает такое сетевое устройство как маршрутизатор.

В пределах семантики иерархического представления модели OSI Сетевой уровень отвечает на запросы обслуживания от Транспортного уровня и направляет запросы обслуживания на Канальный уровень.

Пример: IP/IPv4/IPv6 (Internet Protocol), IPX (Internetwork Packet Exchange, протокол межсетевого обмена), X.25 (частично этот протокол реализован на уровне 2) CLNP (сетевой протокол без организации соединений), IPsec (Internet Protocol Security), ICMP (Internet Control Message Protocol), RIP (Routing Information Protocol), OSPF (Open Shortest Path First), ARP (Address Resolution Protocol).

**Канальный уровень (англ. *Data Link layer*)**

Канальный уровень (англ. Data Link layer) — уровень сетевой модели OSI, который предназначен для обеспечения взаимодействия сетей на физическом уровне и контроля за ошибками, которые могут возникнуть. Полученные с физического уровня данные он упаковывает во фреймы, проверяет на целостность, если нужно исправляет ошибки (посылает повторный запрос поврежденного кадра) и отправляет на сетевой уровень. Канальный уровень может взаимодействовать с одним или несколькими физическими уровнями, контролируя и управляя этим взаимодействием. Спецификация IEEE 802 разделяет этот уровень на 2 подуровня — MAC (Media Access Control) регулирует доступ к разделяемой физической среде, LLC (Logical Link Control) обеспечивает обслуживание сетевого уровня.

На этом уровне работают: коммутаторы, мосты.

В программировании этот уровень представляет драйвер сетевой платы, в операционных системах имеется программный интерфейс взаимодействия канального и сетевого уровней между собой, это не новый уровень, а просто реализация модели для конкретной ОС. Примеры таких интерфейсов: ODI, NDIS

**Физический уровень (англ. *Physical layer*)**

Самый нижний уровень модели предназначен непосредственно для передачи потока данных. Осуществляет передачу электрических или оптических сигналов в кабель или в радиоэфир и, соответственно, их приём и преобразование в биты данных в соответствии с методами кодирования цифровых сигналов. Другими словами, осуществляет интерфейс между сетевым носителем и сетевым устройством.

На этом уровне работают концентраторы (хабы), повторители (ретрансляторы) сигнала и медиаконверторы.

Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом. К физическому уровню относятся физические, электрические и механические интерфейсы между двумя системами. Физический уровень определяет такие свойства среды сети передачи данных как оптоволокно, витая пара, коаксиальный кабель, спутниковый канал передач данных и т. п. Стандартными типами сетевых интерфейсов, относящимися к физическому уровню, являются: V.35, RS-232C, RS-485, RJ-11, RJ-45, разъемы AUI и BNC.

Протоколы: IRDA, USB, EIA RS-232, EIA-422, EIA-423, RS-449, RS-485, Ethernet (включая 10BASE-T, 10BASE2, 10BASE5, 100BASE-TX, 100BASE-FX, 100BASE-T, 1000BASE-T, 1000BASE-SX и другие), 802.11Wi-Fi, DSL, ISDN, SONET/SDH, GSM Um radio interface, IEEE 802.15, ITU и ITU-T, Firewire, TransferJet, Etherloop, ARINC 818, G.hn/G.9960.

Базовые топологии компьютерных локальных сетей

Локальная сеть представляет собой сборище некоторого количества компьютеров, которые расположены на небольшом расстоянии друг от друга. Но в данной лекции мы будет рассматривать основные базовые топологии сети. Давайте для начала разберемся, что мы будем понимать под этим.

И так, под топологией компьютерных сетей мы будем понимать физическое расположение компьютеров относительно друг друга в сети и способу их соединения между собой. Так же вы должны понимать, что топология локальных сетей всегда относиться именно к локальным сетям (для запоминания буду указывать данный аспект в названии). Просто в глобальных сетях базовая топология сети не важна, так как мы знаем, что сети с коммутацией пакетов передают данные по различным маршрутам, да и базовые топология сети (глобальной) скрыта от наших глаз.

Топология компьютерных локальных сетей определяет такие параметры как:

* надежность работы
* возможность расширения сети
* управление обменом данных
* выбор кабельной линии связи

С математической точки зрения под топологией локальных сетей подразумевается граф, где вершинами является персональные компьютеры и различное коммуникационное оборудование, а ребрами являются как раз таки виды физической связи устройств. Но у нас не математика, поэтому графы в топологии компьютерных локальных сетей называются рабочими станциями, о которых мы говорили на ранних лекциях.

***Топология сети общая шина***

На рисунке все компьютеры подключены к общей шине или магистрали (провод, который соединяет все ПК в одну локальную сеть). Главной отличительной чертой топологии сетей общая шина является равноправие всех рабочих станций, т.е. компьютеров, а поддерживаемым видом связи является полудуплексная связь (т.е. поочередным обмен данными по одному и тому же каналу передачи данных). Давайте не заходить вглубь, а сразу перейдем к достоинствам и недостаткам топологии сети общая шина.

 *Достоинства топологии сети общая шина*

* надежность сети очень высокая

Очевидно, что при выходе из стоя любого компьютера, основная линия связи останется цела и сможет легко функционировать.

* возможность полноценной передачи информации

Данный пункт подразумевает, что происходит передача пакетов абсолютно всем подсоединенным компьютерам, т.е. не надо множество раз отправлять данные на каждый компьютер в сети.

* легкая расширяемость

Для увеличения количества компьютеров в сети достаточно просто подключить его к нашей общей шине и вуаля. Т.е. даже ребенок сможет с данной технологией совладать.

* легкость в управлении

Контроль над сетью так же не является особо сложным элементом в топологии сети общая шина, поэтому ставим очередной плюс.

*Недостатки топологии сети общая шина*

Мы с вами познакомились с общими принципами топологии сети общая шина и с преимуществами. Теперь можно и поговорить о недостатках данной топологии.

* обрыв шины

Если выходит из строя компьютера (рабочая станция), то это поправимо, а вот обрыв шины (главного проводка, на котором и держится вся технология) ведет к неминуемой гибели всей сети.

* сложность нахождения поломки

У нас абсолютно все сетевые адаптеры находятся в рабочем состоянии, поэтому процесс нахождения неисправного сетевого оборудования на нише является очень трудным.

* сложность сетевой кары

Сетевая карта при топологии сети общая шина устроена сложнее, так как ей приходится дополнительно бороться с различными коллизиями, возникающими в сети.

* Терминаторы. Чтобы предотвратить отражение электрических сигналов, на каждом конце кабеля устанавливают терминаторы (terminators), поглощающие эти сигналы. Все концы сетевого кабеля должны быть к чему-нибудь подключены, например к компьютеру или к баррел-коннектору — для увеличения длины кабеля. К любому свободному — неподключенному — концу кабеля должен быть подсоединен терминатор, чтобы предотвратить отражение электрических сигналов..

чтобы не возникало отражений, обязательно на концах кабеля должны присутствовать терминаторы, что само по себе несет дополнительную трудность.

Вот мы и рассмотрели основные достоинства и недостатки топологии сети - общая шина. Я могу сказать, что именно данную топологию используют во многих школах, так как она действительно не требует особых умений и специальных знаний в области телекоммуникационных систем. Далее мы познакомимся с двумя оставшимися топологиями и так же рассмотрим их преимущества и недостатки.

***Топология сети звезда***

Мы уже с вами рассматривали иерархическую структуру построения, так вот топология сети звезда является как бы частным случаем данной структуры. Т.е. в центре стоящее устройство выполняет функции коммутатора (switch): соединяет компьютеры друг с другому, естественно после подачи сигнала одним из них.

Как мы видим из рисунка, центральным звеном в топологии сети звезда выступает концентратор, но на данном месте может находиться и еще один компьютер. Надо заметить, что у главной части топологии сети звезда состав оборудования более мощный и, конечно же, более сложный, т.к. все функции выполняются именно на нем.

Если центральным звеном в топологии сети звезда является концентратор, то происходит передача информации сразу всех рабочим станциям локальной сети, т.е. кроме отсутствия магистрали у нас будет топология сети общая шина. А вот с компьютером в центре все происходит централизованно и различные конфликты, при передаче данных от одной рабочей станции к другой, становятся просто неосуществимы. И ни о каком равноправии, как в предвидущей топологии говорить не приходится, ведь именно главный элемент управляет и контролирует все происходящее.

Давайте поговорим о надежности топологии сети звезда. Выход из рабочего состояния одного из звеньев нашей топологии ни как не повлияет на работу всего механизма звезды, а вот выход из строя центрального компонента убивает сразу все локальную сеть. Поэтому необходимо уделять самое пристальное внимание именно центральной части топологии локальной сети звезда.

 *Положительные стороны топологии сети звезда*

* понятная структура
* простой контроль за надежностью

*Проблемы топологии сети звезда*

Мы изучаем в каждой топологии сети ее проблемы. Но зачем это нужно? А дело все в том, что для каждой определенной цели существуют свой набор предпочтений, которые являются наиболее важными. Т.е. и вы можете, исходя из преимуществ и недостатков, определить какая же топология вам наиболее подходит. А теперь давайте рассмотрим основные проблемы, которые существуют у топологии сети звезда:

* ограничение количества узлов

В топологии сети звезда у главного звена может быть 8 или 16 портов, поэтому множество компьютеров подключить невозможно. Но сразу замечу, что данная проблема решается путем подсоединения дополнительного центрального узла:

* затухание сигналов

Надо отметить, что проблему затухания сигналов проще решить именно в топологии сети звезда, чем в общей шине, так как центральный элемент передает сигналы с одинаковой амплитудой, что логично.

* использование не по max канала связи
* сложность реализации

Вся сложность в топологии сети звезда ложиться на главное звено, в котором очень сложно реализовать сразу кодовое и скоростное преобразования, когда у нас будут разнородные сети.

***Топология сети кольцо***

Давайте начнем с нашей топологией общая шина, вот она на рисунке:

Мы просто соединяем наши концы магистрали друг с другом. И вот что у нас выходит (уж как смог нарисовать):

На рисунке уже изображена топология сети кольцо. Т.е. теперь у нас каждый персональный компьютер подсоединен к соседним компьютерам. И, соответственно, в топологии сети кольцо передавать данные он может только переднему соседу, а получать только от заднего соседа. Этим мы экономим общую рабочую силу, так как в сети при приеме и при передаче работает только один приемник и один передатчик.

Топология сети кольцо отличается от остальных тем, что выполняет функции повторителя (происходит ретрансляция приходящего сигнала на каждой рабочей станции). От звезды наша топология сети кольцо отличается еще и тем, что нет явно выраженного центрального элемента, хотя один компьютер и является управляющим в нашей локальной сети.

Так же из-за отсутствия центрального элемента и соединенной магистрали, топология сети кольцо является более устойчивой, чем пройденные топологии. И свободно можно управляться с большим объемом передаваемых данных, ведь у нас нет ни каких промежуточных звеньев цепи, да и не могут появиться различные коллизии в нашей локальной сети.

К сожалению, выход хотя бы одного персонального компьютера из строя рушит полностью всю нашу топологию сети кольцо. Для обеспечения дополнительной надежности прокладывают вторую линию передачи данных, параллельную основной. Это способствует передачи данных в разные стороны:

В топологии сети кольцо данное решение в разы увеличивает надежность сети, и происходит увеличение передачи данных. Вот мы и рассмотрели все основные используемые топологии сетей. Надо заметить, что существуют и еще их разновидности, которые представляют собой просто смесь из пройденных топологий. Например, топологию сети кольцо частенько в фирмах применяют совместно с топологией сети звезда. Т.е. каждый отдел является кольцом, от которого отходит магистраль к главному компьютеру, образуя звезду.

Не случайно мы рассмотрели все три базовые топологии компьютерных локальных сетей и не выделили лучшей. Это сделано из-за того, что в каждом конкретном случае может подходить конкретная топология. Это может быть связанно и со структурой сети, и с денежными возможностями компании и множества других причин.

 Формат протокола стандарта сети Ethernet и принцип работы

**Ethernét** (*этернет,* от лат*. aether —* эфир) — пакетная технология компьютерных сетей, преимущественно локальных.

Стандарты Ethernet определяют проводные соединения и электрические сигналы на физическом уровне, формат кадров и протоколы управления доступом к среде — на канальном уровне модели OSI. Ethernet в основном описывается стандартами IEEE группы 802.3. Ethernet стал самой распространённой технологией ЛВС в середине 90-х годов прошлого века, вытеснив такие устаревшие технологии, как Arcnet, FDDI и Token ring.

В сетевых технологиях, различают такие понятия как фрейм (frame) и пакет (packet).

Новички сетевых технологий, часто делают ошибки в использовании этих терминов и считают что эти термины являются синонимами, но это не так.

Давайте определимся, что же называют фреймами, а что называют пакетами.

Фреймами называют некоторое число байт, которые содержат в себе заголовк Layer 2 OSI и концевик, вместе с инкапсулированными данными (в инкапсулированных данных обычно содержатся так же другие заголовки, других уровней).

Пакетами же часто описывают Layer 3 заголовок вместе с данными. (так же инкапсулированы могут заголовки вышестоящих уровней), но уже без заголовка Layer 2 и концевика (trailer).

Используя знания, полученные в предыдущих статьях, мы знаем, что hub это устройство первого уровня (то есть устройство не знает ни о какой информации, оно не знает о Layer 2 заголовках, и тем более уж о Layer 3).

Но, в то же время, Switch обычно это Layer 2 устройство (то есть оно понимает заголовок Layer 2 Header) и исходя из этого может делать некоторые действия (например брать MAC адрес получателя, искать к какому порту этот MAC-адрес привязан и отправлять фрейм только туда и никуда больше.).

Так же существуют и Layer 3 коммутаторы.

Итак, спецификация Ethernet. Давайте поговорим о ней. Какие они были, какие они сейчас.

Первым основателем Ethernet спецификации стала такая компания как DIX , на самом деле это группа компаний: Digital Equipment Corp, Intel , Xerox.

В начале 1980х годах, IEEE стандартизировала технологию Ethernet , эта технология разделялась на две части:

802.3 Media Access Control (MAC)

802.2 Logical Link Control (LLC)

Существует несколько версий Ethernet фрейма, давайте рассмотрим их.

1. Ethernet DIX.

 2. IEEE Ethernet 802.3.

1. IEEE 802.3 с SNAP заголовком.

Теперь разберем поля поподробнее.

1. Preamble – преамбула, существует во всех версиях Ethernet кадра. Но есть некоторые отличия.

Эти отличия есть между DIX версией и остальными.

В DIX версии, это поле занимало 8 байт.

Вообще, что такое преамбула вообще? Это некая совокупность 0 и 1, которая используется для синхронизации. То есть говорит ресиверу, что будет принят ethernet кадр.

В DIX преамбула была 8 байт, семь первых байтов содержало последовательность 10101010 и так семь раз (7 байт), последний 8-ой байт выглядел так: 10101011.

В 802.3 преамбула стала 7 байт, которые так содержало последовательность 10101010 (7 раз, 7 байт) и было добавлено еще одно поле, которое назвали SD (Start of Frame Delimiter ), что означает : начало ethernet кадра.

Собственно тоже самое что и в DIX реализации, только выделено дополнительное поле. Вместо одного как в DIX’е.

2. Destination address – адрес получателя. MAC адрес. – 6 байт.

3. Source address – адрес отправителя. MAC адрес. – 6 байт.

4. Length – длина фрейма. Это поле указывает на размер фрейма целиком, для того, чтоб получатель мог “предсказать” окончание пакета. Размер поля 2 байта.

5. Data – непосредственно сами данные, их размер может варьироваться от 46 до 1500 байт.

6. FCS – проверка целостности фрейма.

Эти поля относятся к первой части 802.3 Ethernet – MAC.

Так же присутствует как мы помним и вторая часть LLC, давайте рассмотрим ее поля.

DSAP – Destination Service Access Point. 1 байтовое поле. Это точка доступа к сервису системы получателя, которая указывает на то, в каком месте системы получателя буферов памяти следует разместить данные фрейма.

SSAP – Source Service Access Point – так же 1 байтовое поле. Это точка доступа к сервису системы отправителя, которая указывает на то, в каком месте системы отправителя буферов памяти следует разместить данные фрейма.

Control. Управление. Размер поля 1-2 байта. Это поле указывает на тип сервиса, который необходим для данных. В зависимости от того, какой сервис нужно предоставить, поле может быть как 1 так и 2 байта.

Заголовок SNAP – занимает 5 байт. Состоит из двух полей – OUI и TYPE.

При приеме данных, приемник должен знать, какой из сетевых протоколов должен получить входящие данные (например, IP). Для этого и предназначено набор этих полей SNAP – Sub Network Access Protocol (протокол доступа к подсетям).

OUI – Код организации, 3 байта. Идентификатор организации или производителя. Совпадает с первыми 3-мя байтами MAC адреса отправителя.

TYPE – Локальный код. Поле длиной 2 байта. Функцианально это тоже самое что и Ethertype в заголовке Ethernet II.

Как же может устройство определить, какой тип ethernet кадра принимается? Ведь существует DIX формат (Ethernet II), 802.3 и 802\_3 с SNAP ?

Все очень просто. Давайте рассмотрим алгоритм определения.

1. Устройство получает фрейм. Смотрит на поле Lenght/Type (помним, оно занимает 2 байта). Если значение больше чем 1518 байт (размер всего фрейма с заголовками), то это уже не Ethernet II , а 802.3 или 802.3 SNAP, потому как только в Ethernet II указывается размер в указанном поле.

2. Допустим Lenght/Type у нас больше 1518 (0×5FE).

Здесь нам нужно определить, какой фрейм 802.3 или 802.3 SNAP. Это делается на основе заголовка LLC (802.2), таких как DSAP,SSAP и SNAP. Заметим, что SNAP это расширение заголовков DSAP и SSAP (Сервисов стало настолько много, что в 1 байте не удавалось закодировать тот или иной сервис и пришлось создавать еще одну реализацию, которая называется 802.3 SNAP).

SSAP и DSAP обычно принимают одно и тоже значение. Поле Control принимает обычно значение 0×03, что означает, что нет необходимости устанавливать соединение на канальном уровне (Layer 2).

3. И все же, как определить какой формат Ethernet передается, 802.3 или 802.3 SNAP?

Если передается кадр с SNAP, то значение первого и второго байта данных (по сути это наши SSAP и DSAP) равны 0xAA, а третьего (по сути нашего Control) равняется 0×03.

Вот такой алгоритм работает при том, как определить какой формат кадра передается на приемник.

По формату кадров пока все.

Сейчас немного поговорим о адресации на канальном уровне, так же называемой Mac – адресацией.

На канальном уровне, адресация проходит по MAC адресам (помните, когда рассматривали ethernet кадр, первые поля были Destination Address и Source Address, которые занимали 6 байт). Эти адреса имеют 48 битный формат и записываются в 16-ом виде.

Тут стоит отметить тот факт, что, существуют юникастовые рассылки (грубо говоря точка-точка), а существуют множественные рассылки (от одного к нескольким). Это определяется по первому биту MAC адреса, если этот бит равен 1, то это значит что осуществляется множественная рассылка (например широковещательная рассылка, такой адрес имеет все единицы), если первый бит равен “0″, юникастовая рассылка.

Mac адрес состоит как бы из двух частей. Первые три байта прикреплены к той или иной компании, которая производит сетевые устройства, тоесть например устройства Cisco имеет определнные 3 байта одинаковые (некоторые компании имеют не один такой адрес, а целый пул адресов), а вторые 3 байта, это непосредственно уникальный адрес сетевого устройства.

Поля имеют следующие назначения:

* Преамбула: 7 байт, каждый из которых представляет чередование единиц и нулей 10101010. Преамбула позволяет установить битовую синхронизацию на приемной стороне.
* Ограничитель начала кадра (SFD, start frame delimiter): 1 байт, последовательность 10101011, указывает, что далее последуют информационные поля кадра. Этот байт можно относить к преамбуле.
* Адрес назначения (DA, destination address): 6 байт, указывает MAC-адрес станции (MAC-адреса станций), для которой (которых) предназначен этот кадр. Это может быть единственный физический адрес (unicast), групповой адрес (multicast) или широковещательный адрес (broadcast).
* Адрес отправителя (SA, source address): 6 байт, указывает MAC-адрес станции, которая посылает кадр.
* Поле типа или длины кадра (T or L, type or length): 2 байта. Существуют два базовых формата кадра Ethernet (в английской терминологии raw formats - сырые форматы) - Ethernet\_II и IEEE 802.3 (рис.1), причем различное назначение у них имеет именно рассматриваемое поле. Для кадра Ethernet\_II в этом поле содержится информация о типе кадра. Ниже приведены значения в шестнадцатеричной системе этого поля для некоторых распространенных сетевых протоколов: 0x0800 для IP, 0x0806 для ARP, 0x809B для AppleTalk, 0x0600 для XNS, и 0x8137 для IPX/SPX. С указанием в этом поле конкретного значения (одного из перечисленных) кадр приобретает реальный формат, и в таком формате кадр уже может распространяться по сети.

Для кадра IEEE 802.3 в этом поле содержится выраженный в байтах размер следующего поля - поля данных (LLC Data). Если эта цифра приводит к общей длине кадра меньше 64 байт, то за полем LLC Data добавляется поле Pad. Для протокола более высокого уровня не возникает путаницы с определением типа кадра, так как для кадра IEEE 802.3 значение этого поля не может быть больше 1500 (0x05DC). По этому, в одной сети могут свободно сосуществовать оба формата кадров, более того один сетевой адаптер может взаимодействовать с обоими типами посредством стека протоколов.

* Данные (LLC Data): поле данных, которое обрабатывается подуровнем LLC. Сам по себе кадр IEEE 802.3 еще не окончательный. В зависимости от значений первых нескольких байт этого поля, могут быть три окончательных формата этого кадра IEEE 802.3:
* Ethernet\_802.3 (не стандартный, в настоящее время устаревающий формат, используемый Novell) - первые два байта LLC Data равны 0xFFFF;
* Ethernet\_SNAP (стандартный IEEE 802.2 SNAP формат, которому отдается наибольшее предпочтение в современных сетях, особенно для протокола TCP/IP) - первый байт LLC Data равен 0xAA;
* Ethernet\_802.2 (стандартный IEEE 802.2 формат, взят на вооружение Novell в NetWare 4.0) - первый байт LLC Data не равен ни 0xFF (11111111), ни 0xAA (10101010).
* Дополнительное поле (pad - наполнитель) - заполняется только в том случае, когда поле данных невелико, с целью удлинения длины кадра до минимального размера 64 байта -преамбула не учитывается. Ограничение снизу на минимальную длину кадра необходимо для правильного разрешения коллизий.
* Контрольная последовательность кадра (FCS, frame check sequence): 4-х байтовое поле, в котором указывается контрольная сумма, вычисленная с использованием циклического избыточного кода по полям кадра за исключением преамбулы, SDF и FCS.